INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)” . If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in “sectioning” the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from “photographs” if essential to the understanding of the dissertation. Silver prints of “photographs” may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeib Road
Ann Arbor, Michigan 48106
RULE, Ronald Gordon, 1946-
THE DYNAMIC SCHEDULING APPROACH TO AUTOMATED VEHICLE MACROSCOPIC CONTROL.

The Ohio State University, Ph.D., 1973
Engineering, system science

University Microfilms, A XEROX Company, Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED.
THE DYNAMIC SCHEDULING APPROACH
TO AUTOMATED VEHICLE
MACROSCOPIC CONTROL

DISSERTATION
Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By

** ** **

The Ohio State University
January, 1973

Approved by

Adviser
Department of Electrical Engineering
ACKNOWLEDGEMENTS

The author wishes to first thank his wife, Kathleen, for her loving help throughout this entire study. She was not only a real encouragement during the research phase of this effort, but she was of continual assistance in the many tasks of manuscript preparation — including the preparation of illustrations, typing, and editing.

The author also gratefully acknowledges the many constructive criticisms and useful suggestions made by his adviser, Professor Robert E. Fenton. One will always be indebted to Professor Fenton for his great influence on the author's professional development.

The other members of the author's examination committee—Professor Gordon M. Clark, Professor Robert J. Mayhan, and Professor Karl W. Olson — have been most helpful with their respective suggestions regarding the dissertation.

The author is also grateful to Diane Siekierski, Bill Warner, and Henry Pagean for their excellent typing, programming, and photography efforts, respectively.

The author also wishes to thank Dr. Robert Herman for making it possible for the author to complete the appendices while employed at the General Motors Research Laboratories.

Importantly, the author acknowledges the support of the Federal Highway Administration and the Ohio Department of Highways.
VITA

March 25, 1946

Born - Columbus, Ohio

1966 - 1970

Lab Aide, The Ohio State University Research Foundation, Columbus, Ohio

1970

Bachelor of Science in Electrical Engineering, The Ohio State University, Columbus, Ohio

1970

Master of Science in Electrical Engineering, The Ohio State University, Columbus, Ohio

1970 - 1972

Research Associate, The Ohio State University, Columbus, Ohio

PUBLICATIONS


TECHNICAL REPORTS


FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Communication Systems. Professor Claude E. Warren

Studies in Computer Systems. Professor Kenneth J. Breeding

Studies in Control Systems. Professor Robert E. Fenton

Minor Fields: System Simulation. Professor Gordon M. Clark

Multivariate Statistical Analysis. Professor Peter O. Anderson
# TABLE OF CONTENTS

## ACKNOWLEDGMENTS

## VITA

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>A.</td>
<td>Introduction</td>
</tr>
<tr>
<td>B.</td>
<td>Proposed Solutions</td>
</tr>
<tr>
<td>C.</td>
<td>Automated Vehicle Systems</td>
</tr>
<tr>
<td>D.</td>
<td>Synopsis of the Dissertation</td>
</tr>
<tr>
<td>II.</td>
<td>MACROSCOPIC CONTROL</td>
</tr>
<tr>
<td>A.</td>
<td>Introduction</td>
</tr>
<tr>
<td>B.</td>
<td>The Problem</td>
</tr>
<tr>
<td>C.</td>
<td>An Algorithm Taxonomy</td>
</tr>
<tr>
<td>1.</td>
<td>Position Reference Classification</td>
</tr>
<tr>
<td>2.</td>
<td>Pre- and Post-Dispatch Guideway Allocation</td>
</tr>
<tr>
<td>D.</td>
<td>Algorithm Illustrations</td>
</tr>
<tr>
<td>1.</td>
<td>Introduction</td>
</tr>
<tr>
<td>2.</td>
<td>Examples of Predisposition Algorithms</td>
</tr>
<tr>
<td>3.</td>
<td>An Example of Deferred-Disposition</td>
</tr>
<tr>
<td>E.</td>
<td>The Selection of an Approach</td>
</tr>
<tr>
<td>Chapter</td>
<td>TABLE OF CONTENTS (Continued)</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>III. THE DYNAMIC SCHEDULING APPROACH</td>
<td></td>
</tr>
<tr>
<td>A. A generalized Dual-Mode Network</td>
<td></td>
</tr>
<tr>
<td>B. Interchange Geometrics and Flow Potential</td>
<td></td>
</tr>
<tr>
<td>C. A Dynamic Scheduling Philosophy</td>
<td></td>
</tr>
<tr>
<td>1. Controller Architecture</td>
<td></td>
</tr>
<tr>
<td>2. The Interchange Controller</td>
<td></td>
</tr>
<tr>
<td>3. Maneuvering Trajectories</td>
<td></td>
</tr>
<tr>
<td>D. Interchange Clusters</td>
<td></td>
</tr>
<tr>
<td>IV. INTERCHANGE OPERATION</td>
<td></td>
</tr>
<tr>
<td>A. Analytical Queuing Models</td>
<td></td>
</tr>
<tr>
<td>B. Simulation Models</td>
<td></td>
</tr>
<tr>
<td>1. Maneuvering Section Operation</td>
<td></td>
</tr>
<tr>
<td>2. Interchange Models</td>
<td></td>
</tr>
<tr>
<td>3. Experimental Interchange Studies</td>
<td></td>
</tr>
<tr>
<td>C. Interchange Clusters</td>
<td></td>
</tr>
<tr>
<td>1. Cluster Models</td>
<td></td>
</tr>
<tr>
<td>2. Cluster Model Experiments</td>
<td></td>
</tr>
<tr>
<td>V. CORRIDOR NETWORKS</td>
<td></td>
</tr>
<tr>
<td>A. Alternative Configurations</td>
<td></td>
</tr>
<tr>
<td>1. LINE 1 Network</td>
<td></td>
</tr>
<tr>
<td>2. LINE 2 Network</td>
<td></td>
</tr>
<tr>
<td>3. LINE 3 Network</td>
<td></td>
</tr>
<tr>
<td>B. Corridor Models</td>
<td></td>
</tr>
<tr>
<td>1. LINE 1 Model</td>
<td></td>
</tr>
<tr>
<td>2. LINE 2 Model</td>
<td></td>
</tr>
<tr>
<td>3. LINE 3 Model</td>
<td></td>
</tr>
<tr>
<td>C. A Comparative Simulation Study</td>
<td></td>
</tr>
<tr>
<td>1. The Model Experiments</td>
<td></td>
</tr>
<tr>
<td>2. Corridor Simulation Results</td>
<td></td>
</tr>
<tr>
<td>D. Model Utility</td>
<td></td>
</tr>
<tr>
<td>VI. LOOP NETWORKS</td>
<td></td>
</tr>
<tr>
<td>A. Alternative Configurations</td>
<td></td>
</tr>
<tr>
<td>B. Loop Models</td>
<td></td>
</tr>
<tr>
<td>1. LOOP 1 Model</td>
<td></td>
</tr>
<tr>
<td>2. LOOP 2 Model</td>
<td></td>
</tr>
<tr>
<td>VII. URBAN NETWORKS</td>
<td></td>
</tr>
<tr>
<td>A. Introduction</td>
<td></td>
</tr>
<tr>
<td>B. Routing Policies</td>
<td></td>
</tr>
<tr>
<td>C. The Urban Network Model</td>
<td></td>
</tr>
<tr>
<td>D. Model Experimentation</td>
<td></td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>VIII. SUMMARY AND CONCLUSIONS</td>
<td>180</td>
</tr>
<tr>
<td>Appendix</td>
<td></td>
</tr>
<tr>
<td>I. THE DEFERRED DISPOSITION TRANSPORT</td>
<td>189</td>
</tr>
<tr>
<td>SIMULATOR (DDTS) USER'S MANUAL</td>
<td></td>
</tr>
<tr>
<td>II. THE DDTS INTERNAL LOGIC MANUAL</td>
<td>218</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>266</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Teletype Conversation with DDTS</td>
<td>198</td>
</tr>
<tr>
<td>2. DDTS Messages and the Required Responses</td>
<td>201</td>
</tr>
<tr>
<td>3. RUN-time Commands</td>
<td>208</td>
</tr>
<tr>
<td>4. Simulation Errors</td>
<td>209</td>
</tr>
<tr>
<td>5. DDTS Routines (Alphabetical Listing)</td>
<td>237</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>Macroscopic-Control-Algorithm Taxonomy Tree.</td>
<td>13</td>
</tr>
<tr>
<td>2.</td>
<td>A simple network and its origin-destination paths.</td>
<td>18</td>
</tr>
<tr>
<td>3.</td>
<td>An example of pre-demand guideway space allocation.</td>
<td>20</td>
</tr>
<tr>
<td>4.</td>
<td>An example of post-demand guideway space allocation.</td>
<td>23</td>
</tr>
<tr>
<td>5.</td>
<td>An example of deferred disposition of guideway space.</td>
<td>25</td>
</tr>
<tr>
<td>6.</td>
<td>A representative automated guideway system.</td>
<td>31</td>
</tr>
<tr>
<td>7.</td>
<td>A cloverleaf interchange configuration.</td>
<td>33</td>
</tr>
<tr>
<td>8.</td>
<td>Representative paths through a cloverleaf interchange.</td>
<td>34</td>
</tr>
<tr>
<td>9.</td>
<td>A 4-way directional interchange (multiple bridge structures).</td>
<td>37</td>
</tr>
<tr>
<td>10.</td>
<td>A 4-way directional interchange (four-level bridge structure).</td>
<td>38</td>
</tr>
<tr>
<td>11.</td>
<td>Representative paths through the directional interchanges.</td>
<td>40</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES (continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12. A distributor-collector interchange model.</td>
<td>41</td>
</tr>
<tr>
<td>13. A 3-way and 4-way interchange, and a representative entrance-exit facility.</td>
<td>42</td>
</tr>
<tr>
<td>14. A representative portion of a guideway network.</td>
<td>44</td>
</tr>
<tr>
<td>15. The controller architecture.</td>
<td>44</td>
</tr>
<tr>
<td>16. The realm of interchange controller jurisdiction.</td>
<td>47</td>
</tr>
<tr>
<td>17. Dynamic &quot;path&quot; scheduling.</td>
<td>48</td>
</tr>
<tr>
<td>18. Representative maneuvering trajectories.</td>
<td>50</td>
</tr>
<tr>
<td>19. A maneuvering section in the &quot;state&quot; of maximum vehicle storage.</td>
<td>54</td>
</tr>
<tr>
<td>20. Interchange &quot;cluster&quot; configurations.</td>
<td>57</td>
</tr>
<tr>
<td>21. A 2-way interchange queuing model.</td>
<td>59</td>
</tr>
<tr>
<td>22. A 3-way interchange queuing model.</td>
<td>65</td>
</tr>
<tr>
<td>23. Maximum average per-lane flow density versus demand balance (γ).</td>
<td>67</td>
</tr>
<tr>
<td>24. Basic simulation-model elements.</td>
<td>72</td>
</tr>
<tr>
<td>25. A 2-way interchange and its downstream neighbors. A 2-way simulation model is also shown.</td>
<td>77</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>26</td>
<td>The required &quot;information flow&quot; paths.</td>
</tr>
<tr>
<td>27</td>
<td>A 3-way interchange simulation model.</td>
</tr>
<tr>
<td>28</td>
<td>A 4-way interchange simulation model.</td>
</tr>
<tr>
<td>29</td>
<td>An entrance-exit interchange-facility simulation model.</td>
</tr>
<tr>
<td>30</td>
<td>Experimentally obtained flow-density characteristics for a 3-way interchange.</td>
</tr>
<tr>
<td>31</td>
<td>Experimentally obtained delay characteristics for a 3-way interchange.</td>
</tr>
<tr>
<td>32</td>
<td>Experimentally obtained delay characteristics for a 3-way interchange (log scale).</td>
</tr>
<tr>
<td>33</td>
<td>A 3-way interchange-cluster model.</td>
</tr>
<tr>
<td>34</td>
<td>A 4-way interchange-cluster model.</td>
</tr>
<tr>
<td>35</td>
<td>Three corridor networks.</td>
</tr>
<tr>
<td>36</td>
<td>The LINE 1 Configuration.</td>
</tr>
<tr>
<td>37</td>
<td>The LINE 2 Configuration.</td>
</tr>
<tr>
<td>38</td>
<td>The parallel entrance-exit facility.</td>
</tr>
<tr>
<td>39</td>
<td>The LINE 3 Configuration.</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>40.</td>
<td>The longitudinal and lateral maneuvering areas necessary for lane-changing operations.</td>
</tr>
<tr>
<td>41.</td>
<td>Possible combinations of lane-changing requests for two adjacent lanes.</td>
</tr>
<tr>
<td>42.</td>
<td>The analogy between lane-changing and 2-way interchange operation.</td>
</tr>
<tr>
<td>43.</td>
<td>The LINE 1 Corridor Model.</td>
</tr>
<tr>
<td>44.</td>
<td>The LINE 2 Corridor Model.</td>
</tr>
<tr>
<td>45.</td>
<td>The LINE 3 Corridor Model.</td>
</tr>
<tr>
<td>46.</td>
<td>Average flow rates in the corridor models.</td>
</tr>
<tr>
<td>47.</td>
<td>Estimated origin-destination flow rates.</td>
</tr>
<tr>
<td>48.</td>
<td>Entrance and exit flow rates.</td>
</tr>
<tr>
<td>49.</td>
<td>Entrance rejection rates for the corridor models.</td>
</tr>
<tr>
<td>50.</td>
<td>Average delays (in slot-times) for the Corridor Models.</td>
</tr>
<tr>
<td>51.</td>
<td>Representative loop networks.</td>
</tr>
<tr>
<td>52.</td>
<td>Routing patterns for a 5-Facility LOOP 1 Model.</td>
</tr>
<tr>
<td>53.</td>
<td>Flow densities for the 5-Facility LOOP 1 Model.</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES (Continued)

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>54. The clockwise-loop (model) of the 4-facility LOOP 2 configuration.</td>
<td>136</td>
</tr>
<tr>
<td>55. Average delay versus on-loop flow density for the LOOP 2 Models.</td>
<td>138</td>
</tr>
<tr>
<td>56. Standard-deviation of delay versus on-loop flow density for the LOOP 2 Models.</td>
<td>139</td>
</tr>
<tr>
<td>57. Average delays for various trip lengths in the 4-facility LOOP 2 Model.</td>
<td>140</td>
</tr>
<tr>
<td>58. Standard deviation of delay for various trip lengths in the 4-facility LOOP 2 Model.</td>
<td>141</td>
</tr>
<tr>
<td>59. Average delays for various trip lengths in the 8-facility LOOP 2 Model.</td>
<td>142</td>
</tr>
<tr>
<td>60. Standard deviation of delay for various trip lengths in the 8-facility LOOP 2 Model.</td>
<td>143</td>
</tr>
<tr>
<td>61. Facility I with appropriate additions for modeling egress limitations.</td>
<td>144</td>
</tr>
<tr>
<td>62. Flows at Facility II for case with no egress restrictions.</td>
<td>146</td>
</tr>
<tr>
<td>63. Flows at Facility II for case with an egress restriction at Facility I at t = 3000.</td>
<td>147</td>
</tr>
<tr>
<td>64. A representative urban network.</td>
<td>150</td>
</tr>
<tr>
<td>65. The urban network configuration studied.</td>
<td>151</td>
</tr>
<tr>
<td>Figures</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>66. A simplified representation of the urban network.</td>
<td>152</td>
</tr>
<tr>
<td>67. Some possible paths from A to J, and from G to C.</td>
<td>153</td>
</tr>
<tr>
<td>68. An urban network model.</td>
<td>157</td>
</tr>
<tr>
<td>69. Paths originating at one of the inner loop facilities (A).</td>
<td>158</td>
</tr>
<tr>
<td>70. Paths originating at C and D.</td>
<td>159</td>
</tr>
<tr>
<td>71. Total average flow rate into each interchange.</td>
<td>162</td>
</tr>
<tr>
<td>72. The number-of-completed-trips matrix (case $\lambda = .065$).</td>
<td>164</td>
</tr>
<tr>
<td>73. Flow measurement points in one quadrant of the urban model.</td>
<td>165</td>
</tr>
<tr>
<td>74. Flow rates at Interchange $\alpha$ (case $\lambda = .065$).</td>
<td>166</td>
</tr>
<tr>
<td>75. Flow rates at Interchange $\beta$ (case $\lambda = .065$).</td>
<td>167</td>
</tr>
<tr>
<td>76. Flow rates at Interchange $\gamma$ (case $\lambda = .065$).</td>
<td>168</td>
</tr>
<tr>
<td>77. Flow rates at Interchange $\delta$ (case $\lambda = .065$).</td>
<td>169</td>
</tr>
<tr>
<td>78. Flow rates at Interchange $\epsilon$ (case $\lambda = .065$).</td>
<td>170</td>
</tr>
<tr>
<td>79. The minimum-travel-time matrix (case $\lambda = .065$).</td>
<td>172</td>
</tr>
</tbody>
</table>
LIST OF FIGURES (Continued)

Figures

80. The mean-travel-time matrix (case \( \lambda = .065 \)). 173

81. The max-travel-time matrix (case \( \lambda = .065 \)). 174

82. The standard-deviation-of-travel-time matrix (case \( \lambda = .065 \)). 175

83. Delay statistics for the urban model subject to a balanced demand. 176

84. Average delay as function of the demand level (\( \lambda \)) with trip length as a parameter. 177

85. Standard-deviation of delay as a function of the demand level (\( \lambda \)) with trip length as a parameter. 178

86. Control transfer among modules. 191

87. The complete set of SIMU-level commands. 192

88. Accessing modules directly from the PDP-10 Monitor level. 194

89. An example of a sequence of control transfers among modules. 195

90. Transfer of control via module reentry points. 196

91. Specifications for the LINEl Model. 211
<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>92. Specifications for the LINE2 Model.</td>
<td>212</td>
</tr>
<tr>
<td>93. Specifications for the LINE3 Model.</td>
<td>213</td>
</tr>
<tr>
<td>94. Specifications for the CLUSTER-3 Model.</td>
<td>214</td>
</tr>
<tr>
<td>95. Specifications for the CLUSTER-4 Model.</td>
<td>215</td>
</tr>
<tr>
<td>96. The required O/D matrix form for the LINE1 and LINE2 Models.</td>
<td>216</td>
</tr>
<tr>
<td>97. The required O/D matrix form for the LINE3 Model.</td>
<td>217</td>
</tr>
<tr>
<td>98. The format of vehicle-words.</td>
<td>219</td>
</tr>
<tr>
<td>99. The element-group register format.</td>
<td>222</td>
</tr>
<tr>
<td>100. SIMU Module flow chart.</td>
<td>224</td>
</tr>
<tr>
<td>101 ESTM Module flow chart.</td>
<td>225</td>
</tr>
<tr>
<td>102 BUILD Module flow chart.</td>
<td>226</td>
</tr>
<tr>
<td>103 SETUP Module flow chart.</td>
<td>227</td>
</tr>
<tr>
<td>104 INIT Module flow chart.</td>
<td>228</td>
</tr>
<tr>
<td>Figures</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>105 RUN Module flow chart.</td>
<td>229</td>
</tr>
<tr>
<td>106 STATS Module flow chart.</td>
<td>230</td>
</tr>
<tr>
<td>107 SIMU Module routine nesting.</td>
<td>232</td>
</tr>
<tr>
<td>108 BUILD Module routine nesting.</td>
<td>233</td>
</tr>
<tr>
<td>109 ESTM, SETUP, and INIT Module routine nesting.</td>
<td>234</td>
</tr>
<tr>
<td>110 RUN Module routine nesting.</td>
<td>235</td>
</tr>
<tr>
<td>111 STATS Module routine nesting.</td>
<td>236</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

A. Introduction

The automobile has gained a dominant position for transportation in both urban and inter-city environments. This can be attributed to several factors: first, its unique point-to-point mobility; second, both the privacy it offers and the freedom a traveler has to choose his traveling companions, departure time, route of travel, and, to a certain extent, his pace; and finally, it provides an "at home" environment away from home, plus a certain value as a status symbol.

However, many problems have evolved with the growth of the automotive system of transportation [27] including extremely congested roads, an alarmingly high rate of traffic accidents and fatalities, inefficient land usage, and environmental pollution. Also, the system has not kept pace with user demands that have arisen with population increases, urban sprawl, greater affluence, and more leisure time.

It appears that the driver has become one of the chief problematical elements as traffic densities have increased. This has become especially evident in tasks involving vehicle interactions, e.g., car-following, merging, and lane-changing. Thus, a driver's car-following behavior in vehicle platoons has been found to be severely
inadequate, resulting in the propagation of disturbances and the all too frequent multi-vehicle freeway accident [11], [24]. Also, drivers function poorly in merging onto a limited-access highway -- due in part to their limited ability to detect and track mainstream gaps [13]. Further, it appears that drivers tend to make improper lane-changing and routing decisions in the face of complex multilane interchange configurations and high flow densities [38].

B. Proposed Solutions

There have been many proposed solutions to automotive transportation problems, and these have generally fallen into three categories:

(1) The improvement of the automobile mode of transportation by both aiding the driver, and making vehicle and roadway improvements;

(2) The improvement of mass-transit service (and thereby reduce the demand for automotive travel);

(3) The development of automated vehicle systems for both urban and inter-city use.

Considerable effort has been expended in the first category, e.g., driver aids have been developed for the car-following task [6], [35], for the merging maneuver [7], [9], and for route guidance [22], [38]. Further, changes have been made to both improve the safety and
handling qualities of the automobile, together with interchange layouts, roadway markings and surfaces, and the coverage of urban areas by limited-access highways [1].

The improvement of public mass transit has had a high priority during recent years with one major result being the Bay Area Rapid Transit System serving the San Francisco area. In some areas, where demands are not sufficiently concentrated to allow economical "line-haul" rail service, other means such as improved bus service (e.g., Dial-a-bus) are being planned or utilized.

In most mass-transit systems, however, circuitous routing and intermediate stops are necessary for efficient operation, even though the resulting inconvenience and delays tend to discourage user acceptance. Thus, such systems appear to provide only partial solutions to alleviating automotive transportation problems. In contrast, various systems using small automated vehicles (small in comparison to mass-transit ones) have been proposed — as it is felt that these would retain many of the advantages of the "conventional" automotive system (such as its point-to-point mobility), yet eliminate one of the chief problematical elements — namely, the driver.

C. Automated Vehicle Systems

Three general classes of automated-vehicle systems have been proposed — pallet-vehicle, captive-vehicle, and dual-mode-vehicle systems. A pallet-vehicle system utilizes automated "carriers" upon which conventional automobiles (occupied by passengers) ride [31].
Such a system would provide "personal rapid transit" (a designation that has been given to automated-vehicle systems) over a large portion of the user's trip, yet would avoid most of the inconveniences of mass-transit systems.

In the second class of automated-vehicle systems, "captive-vehicles" would be used. Passengers would board such vehicles at stations where unoccupied vehicles are stored. Although such systems would not provide the point-to-point mobility of the automobile, they are potentially useful in restricted geographical areas such as central business districts, airports, and college campuses. An example is the prototype system presently being installed between the separated campuses of West Virginia University by the U. S. Department of Transportation. This system will be used to help evaluate the feasibility of this type of automated ground transport.

In the third class of systems dual-mode vehicles, which operate manually on conventional roads and automatically on guideways (automated highways), would be utilized. Both this class of systems and pallet-vehicle systems would clearly retain the point-to-point accessibility available to today's automobile traveler.

Many rather general feasibility studies have been undertaken [8], [15], [20], [21], [25], [29], [31], [44], while other studies have dealt specifically with various microscopic and macroscopic control problems associated with automated vehicle systems. Microcontrol is explicitly concerned with individual vehicle position-regulation and -maneuvering while macrocontrol embodies the entire hierarchy of con-
trol necessary for system coordination.

In the former area both theoretical and experimental studies have generally fallen into the following three categories: longitudinal [3-5], [26], [15], lateral [33], and merge control [2], [13-14].

The research reported is specifically directed towards the development of macroscopic control "approaches" (suited to the task of scheduling and routing dual-mode or automated highway vehicles), and a relevant literature survey is contained in the next chapter.

D. Synopsis of the Dissertation

The dissertation is organized as follows. In Chapter II the macroscopic control task is defined, a control algorithm taxonomy presented, and a particular genus is selected for detailed study. The use of a combination of path reservations through interchanges (as each one is approached) and maneuvering-space reservations (for the next interchange) are the sine qua non of this approach.

In Chapter III, a generalized dual-mode network model is presented and the selected approach, "dynamic scheduling," is described in terms of this model. The controller architecture employed incorporates both a central coordinator and local interchange controllers.

In Chapter IV, the operation of local controllers for both interchanges and entrance-exit facilities is modeled. Flow limitations resulting from vehicle-maneuvering interactions are brought into focus by empirical results (obtained from computer realizations of these models).
In Chapter V, the importance of both entrance-exit facility design and multiple-lane usage are studied. This chapter is centered around the comparative study of three designs — each intended to serve a four entrance-exit facility corridor. Of distinct importance is a demonstration of the effectiveness of a simple swerve-maneuver policy wherein each lane spanning the distance between entrance-exit interchanges is dichotomized into longitudinal- and lateral-maneuver areas — longitudinal maneuvering being preparatory to lateral maneuvering.

In Chapter VI, problems associated with network loops are briefly highlighted while in Chapter VII, simulation results for an urban network, having an inner ring, outer ring and four radials, are presented. As this chapter is only an initial step into the realm of large network control, more problems are defined than are solved. A summary and discussion of the main results, the conclusions reached, and ideas for future research are contained in Chapter VIII.

The Appendices provide a brief documentation of the network simulator — as its conception, design, and construction consumed a great portion of the research effort, yet its utility vastly broadened the scope of the macroscopic control area that could be explored.
CHAPTER II
MACROSCOPIC CONTROL

A. Introduction

This chapter contains both a description of the macroscopic control problem and a taxonomy of some of its solutions. Initially, the problem is defined, and observations pertinent to automated-vehicle systems and their operating environment are presented. A taxonomy tree of solutions is constructed and used to classify algorithms used in prior studies, and three classes of solutions are illustrated using a simple network. Finally, the approach, that is to be the focus of this dissertation, is selected.

B. The Problem

The "macroscopic control task" can be described as the search for a procedure suited to the task of allocating space and specifying trajectories for vehicles traversing an automated-vehicle network. This would embody the highest level of system supervision or coordination, and its sine qua non would be the ability to coordinate vehicle movement such that each would be routed from its origin to its destination in an efficient, safe, and conflict-free fashion. It would also be imperative that sufficiently high flow rates be achievable.
Unfortunately the growth in the research area of "macroscopic control of automated ground transportation networks" has been very slow in its approximate decade of existence -- possibly due to the striking lack of conceptual tools or mathematical constructs that are useful in the inventive process of synthesizing such systems. This scarcity can be attributed to the complexity of modeling automated transportation networks, for this involves several more fundamental and yet rather abstract disciplines (as stochastic processes, queuing theory, and network topology).

Consider the following characteristics of, and constraints on, the operation of an automated-vehicle system:

(1) Any such system would consist of a fixed and hence limited set of resources (roadway, interchanges, storage areas, etc.) which a macroscopic controller can allocate to satisfy users' demands.

(2) These demands comprise a non-stationary stochastic process; i.e., the system would be subjected to non-deterministic short- and long-term time-dependent fluctuations in the pattern of trip requests.

(3) Interchanges (that is, areas where vehicle flows interact and are redistributed according to individual vehicle
destinations) are usually flow limiting points within a guideway (roadway) network and hence guideway links interconnecting interchanges may be forced to operate below their maximum capacity.

(4) There is limited space at entrances and/or interior to a network for static or dynamic vehicle storage. Also, interior to a network, there is limited distance in which to maneuver vehicles to resolve merge and lane-changing conflicts.

(5) The guideway network should have multiple lanes or alternate routes through portions of the system where heavy demands are expected. This is essential to the task of diverting vehicles around anomalous blockages -- which should be both expected and planned for (though certainly not welcomed). However, the extent to which the network can be "over-designed" in this manner will obviously be limited by both economic and other constraints (for example, the noneconomic aspects
of land acquisition).

(6) To maximize user utility it seems desirable that the macroscopic control scheme be chosen to minimize both some measure of travel time and its dispersion.\(^1\) Also, it is important to minimize the rejection probability at system entrances. (Rejection can occur both during excessive demand peaks and when anomalous conditions have developed.)

(7) The system should be highly adaptive to user demand. For example, such adaptation would include updating routing patterns to adjust to through-put loads altered by demand shifts.

(8) One most important characteristic of an automated-vehicle system is that it be fault tolerant. This dictates that the macroscopic control scheme incorporate modes of operation that allow for re-routing and re-scheduling of vehicles should any portion of the system become blocked. It must be able to both detour

---

\(^1\) A complete "user utility function" would include other factors as safety, reliability, and cost.
vehicles not having reached an affected area and restart vehicles within same once the anomaly is corrected.

(9) Finally it seems that a hierarchy of subcontrollers has definite advantages over a single-centralized master controller. First, the communications requirements in a hierarchical system would undoubtedly be smaller because subcontrollers could operate somewhat autonomously, and further, the reliability requirements would not be as stringent since localized controllers could switch to completely autonomous modes were communications with higher levels of command temporarily impaired.

The above list represents only a preliminary set of operational characteristics and constraints pertinent to any automated vehicle system; however, it will be useful in the subjective evaluation of several fundamental approaches to the macroscopic control problem.

C. An Algorithm Taxonomy

In this section, several fundamental approaches to macroscopic control are presented by means of a taxonomy tree. Prior re-
search efforts are then fitted into this structure.

As a preface, it is well to point out that macroscopic control research is still in an early stage of development and nothing more than partial control philosophies have been proposed. For this reason, the taxonomy tree presented in Fig. 1. does not include classifications according to such aspects as routing and failure recovery procedures. Also, facets of control, which are unimportant in dual-mode systems are not considered here (such as vehicle inventory procedures which are required in both captive- and pallet-vehicle systems).\(^2\)

1. Position Reference Classification

The first criterion of classification is based on the rather microscopic problem of vehicle position reference. Historically, automated vehicle system planning has generally fallen into either of two classes:

(1) Those utilizing inter-vehicular position referencing (commonly termed "car-following" approaches) and

(2) Those employing a roadway (guideway) based moving position reference (commonly termed the "synchronous," "slot," or "moving-cell" approach).

\(^2\)It is of course recognized that a "dual-mode system" could well include both captive and dual-mode vehicles.
Macroscopic Control Algorithms

Based on inter-vehicular position referencing ("car-following" systems)

Based on a guideway-based moving position reference ("synchronous" systems)

Pre-dispatch guideway allocation ("predisposition" systems)

Post-dispatch guideway allocation ("deferred-disposition" systems)

Cycle reservation systems

Slot reservation

Pre-demand guideway allocation ("prescheduling")

Post-demand guideway allocation ("responsive scheduling")

Pre-merge scheduling

Commonly termed "free scheduling"

Maneuvering-space limitations not handled

Pre-interchange scheduling

Rerouting to handle maneuvering-space limitations

Dynamic reservation of maneuvering space

Fig. 1—Macroscopic-Control-Algorithm Taxonomy Tree.
The first class was initially examined at General Motors Corporation [31], subsequent efforts were by such diverse groups as Road Research Laboratory (Great Britain) [34], Government Mechanical Laboratory (Japan) [18], and the Ohio State University [15]. However, much of this research has been in the area of instrumentation feasibility rather than macroscopic control, and it has since been generally thought that "car-following" systems are most useful in headway protection subsystems and controllers designed for light demand areas (as some intercity links). For this reason this class of systems is not further considered in this study.

Researchers explicitly interested in macroscopic control philosophies, have directed their attention to the second class and generally envisioned a roadway-based moving position reference. This concept had its origins at least as early as 1964 when Cluck [12] likened the automated highway entry process

... to that of a conveyor belt moving at a constant velocity and containing evenly spaced "slots." The vehicles are then placed into these slots.

The concept of synchronous operation was further advanced by The Glideway System MIT Student Project [16] (1965) as evidenced by the statement:

Intervehicle reactions require that the control of any one vehicle be a function of all the other vehicles on the system. The complexity of such a control system may be significantly reduced by the introduction of the concept of constant speed synchronous operation. Synchronous operation may be likened to the operation of a conveyor belt. Every part of a conveyor belt is moving at the same velocity.
Subsequently, Godfrey [17] (1968), TRW Systems, Inc. [39] (1969), Wilkie [42] (1970), Carlson [10] (1971) and others have applied this concept. TRW Systems, besides greatly popularizing the synchronous concept, made the following important generalization concerning the slots:

The vehicle spaces, called slots, represent a period of time rather than a fixed distance, so that they stretch and contract to accommodate the changing speed profile along the guideway.

2. Pre- and Post-Dispatch Guideway Allocation

Returning to the taxonomy tree, shown in Fig. 1, note that the class of synchronous macroscopic controllers have been subdivided according to whether guideway space is allocated before or after vehicle dispatch. (The latter will be loosely defined as the point in time when a vehicle is released from an entrance facility and begins travel towards its destination.) The two subclasses of synchronous systems are:

(1) Pre-dispatch guideway allocation systems

(commonly referred to as "preprogrammed" or "reservation" systems) and

(2) Post-dispatch guideway allocation systems

which will be termed deferred-disposition systems.

2.1 Predisposition Systems

The first class of systems, which will be referred to as "pre-disposition" systems, are by far the most popular of the two, pos-
ibly due to both their conceptual simplicity and early development. Such systems originated (to the best of this author's knowledge) with TRW Systems, Inc. [39], who introduced the concept of allocating guideway space by reserving a "slot" position-time trajectory for a vehicle prior to dispatching that vehicle. This guarantees a conflict-free trajectory (assuming no breakdowns occur) and eliminates the need to maneuver vehicles prior to merges. The more flexible scheme of "cycle" reservation was also introduced by TRW Systems, Inc. [39]. Here, entering vehicles would wait in an entrance queue until a "cycle of contiguous slots" passed, wherein at least one slot were available (unreserved) from the waiting vehicle's origin to its destination. Cycle size (number of slots per cycle) is limited by the availability of space within the network for maneuvering vehicles among moving slots (but within their cycles) to resolve merge conflicts. An apparent advantage of "cycle" over "slot" reservation is that entering vehicles can probably be serviced sooner, as waiting for a cycle "vacancy" is a less stringent requirement than slot acquisition. Note from Fig. 1 that predisposition systems have been subdivided into the aforementioned -- slot reservation systems and cycle reservation systems. A further division of the latter class is presented in section D.

2.2 Deferred-Disposition Systems

In contrast to predisposition systems wherein guideway space would be reserved prior to vehicle dispatch, deferred-disposition systems dynamically allocate guideway space to vehicles as they proceed through the system. Unfortunately very little progress has been made
in the development of such systems, most likely due to the perplexing nature of what one will term the "over-capacity protection problem" — that is, what procedure can one utilize to guarantee that each vehicle on the system can be maneuvered to resolve all merge conflicts?

This problem was alluded to in 1964 by Cluck [12]:

"... special events could on occasion cause the demand to exceed the capacity."

In 1968 Godfrey [17], most clearly defined the problem:

"The imposition of a maximum allowed maneuver will, because of overflows, require extensive network control integration ..."

Of considerable importance are the quantitative estimates of maneuvering-space requirements that Godfrey obtained in his research of the synchronous merge problem.

No major breakthrough in solving this problem had yet come, however, until Munson and Travis [30] of the Aerospace Corporation (in May of 1971) devised a rerouting scheme to circumvent physical maneuvering limitations. They state:

"Because of physical limitations at intersections, the intersection control computer will not always be able to manipulate the traffic streams so that all cars that should turn at an intersection will be able to do so. Those cars which cannot turn will be sent straight through the intersection and rerouted to their destinations at subsequent intersections.

Although this concept is clearly limited to a narrow class of intersections (the one they consider is shown in Fig. 2 (a)), and network configurations (one with multiple routes is necessary), it definitely demonstrates the existence of viable deferred-disposition approaches."
Fig. 2—A simple network and its origin-destination paths.
Before selecting a particular macroscopic control approach for detailed study, both the predisposition and the deferred-disposition approaches are illustrated in the next section.

D. Algorithm Illustrations

1. Introduction

A simple network consisting of the intersection of two single-lane one-way roadways or guideways is displayed in Fig. 2 (a). Vehicles enter at points 1 and 2 and depart at either A or B. A topologically equivalent network is shown in Fig. 2 (b). Here, two paths through the network, 1-A and 2-B, have been defined and these along with 1-B and 2-A comprise all possible paths.

The state of the network can be displayed by indicating the positions of vehicles along their respective paths. By adding time as a second variable, the progression of network state with time is shown in the form of vehicle trajectories. It is especially useful to plot the state at discrete instances in time if synchronous movement of vehicles is prescribed (during all or portions of their trips). If time were incremented in single units of time-headway, then vehicle positions would "fall into slots" at equi-spaced points along each path -- in portions of the network where the vehicles are moving synchronously.

2. Examples of Predisposition Algorithms

2.1 An Example of Pre-Demand Allocation

In Fig. 3 the state of the path, 1-A, defined by the inter-
Fig. 3—An example of pre-demand guideway space allocation.
change in Fig. 2 is plotted\textsuperscript{3} for one type of pre-demand allocating controller and an arbitrary source stream. A vehicle is represented by a letter corresponding to its destination (A or B). As a vehicle progresses at a constant speed, its trajectory is a diagonal line. With pre-demand guideway allocation schemes the allowable trajectories for each path through the network are prescheduled in accordance with historic or current demand trends — more heavily demanded routes (paths) being allocated a larger share of the available trajectories. (In Fig. 3 prescheduled trajectories are represented by variously shaded diagonal sets of blocks.) In this simple example, preschedules for each trip type are arbitrarily shown equally as often. Further the patterns of prescheduling at each entrance are cyclic with a period of two slots — slots being alternately destined to each of the two exits.

An entering vehicle, such as the A-bound vehicle in source stream 1 at slot-time 1 (Fig. 3), is delayed at the "entrance" until a slot, prescheduled to the appropriate destination, appears. In this case the vehicle is delayed by one slot-time (or one time-headway unit) before linking up with the prescribed trajectory to its destination. In contrast, the A-bound vehicle, entering at slot-time 8, travels through the network with no delay.

2.2 An Example of Post-Demand Allocation

In contrast to pre-demand schemes wherein preschedules are used, post-demand guideway allocating schemes generate schedules in

\textsuperscript{3}To completely describe the operation of the simple interchange considered here, the state of all four possible paths would be plotted.
real-time to reflect individual vehicle demands.

In Fig. 4, trajectories are specially scheduled for each vehicle as it arrives in the source stream, and there are no unused scheduled-paths (denoted by diagonal sets of shaded blocks) as in the previous example.

The scheduling discipline used here is order-of-arrival service with priority to entrance 1 over entrance 2. Hence, the A-bound vehicle in the source stream at entrance 1 and slot-time 1 is dispatched prior to another A-bound vehicle found in source stream 2 at the same time (Only the last three slots of the latter vehicle's trip are shown in Fig. 4 since the state of path 2-A is not plotted). However, since service is in order-of-arrival, both of these vehicles are dispatched prior to dispatching the vehicle in the source stream at time 2.

In both of the examples given (predisposition algorithms) the future trajectory of each vehicle is specified prior to its dispatch (departure from its entrance area). In this way a scheduler can assure a conflict-free path for each vehicle by simply detaining it at an entrance storage facility until a path can be reserved.

3. An Example of Deferred-Disposition

In contrast to predisposition systems wherein guideway space is reserved prior to vehicle dispatch, space in deferred-disposition systems is allocated dynamically to each vehicle as it proceeds through the system. This dynamic allocation of space is usually accomplished by allowing vehicles to travel synchronously (in prescribed slots)
Fig. 4—An example of post-demand guideway space allocation.
until they are at points within the system where imminent interchange or merge conflicts must be resolved. Such conflicts are resolved by a maneuvering operation such as the one illustrated in Fig. 5. Note here that a deferred-disposition scheme has been employed and, at least in the example illustrated, none of the entering vehicles are required to wait, although one of the vehicles on the path 1-A (which is the only path illustrated) is required to undertake a delay maneuver, commencing at time 6 and ending at time 13. The maneuvering procedure chosen here is a rather simple one — the vehicle merely tracks a slower moving synchronous reference (traveling at 6/7th normal synchronous speed). (Note also that the maneuvering-vehicle's slot length is foreshortened.) The vehicle tracks the slower reference for 7 slot times to "slip back" one slot with respect to the normal reference. If it were required to delay a vehicle two slots in this example, due to the limited maneuvering space between the diverge and the merge areas, the scheduler would probably accomplish the delay by a two-step procedure — half the delay being accomplished by detaining the vehicle at the entrance facility.

E. The Selection of an Approach

The class of algorithms selected for detailed investigation is defined by the following path through the taxonomy tree (Fig. 1):

(1) A guideway-based moving position reference ("synchronous" systems),

(2) Post-dispatch guideway allocation ("deferred-disposition" systems),
Fig. 5—An example of deferred disposition of guideway space.
(3) Pre-interchange scheduling, and
(4) Dynamic reservation of maneuvering space.

The rationale behind this selection is best described by delineating the considerations involved in the selection of each branch of the taxonomy tree.

The synchronous approach (i.e., using a guideway-based moving position reference) is selected (rather than a "car-following" one) as it would:

(1) greatly simplify vehicle coordination,
(2) possibly reduce computer and communication requirements [29], and
(3) avoid platoon stability problems [4]

(which would be encountered in automated "car-following" systems).

A method of deferred-disposition allocation was selected rather than predisposition allocation as it appears that members of the former

(1) Are more fault tolerant,
(2) Are better able to handle exiting backups,
(3) Can use the guideway for "dynamic storage,"
(4) Have smaller entrance storage requirements,
(5) Are less vulnerable to shutdowns, as a central master scheduler is not required (although a central coordinator may perform non-critical tasks),
(6) Do not require rigid time synchronization over a whole network.

Fault tolerance (in this context) is a relative measure of the capability of a system to adapt to anomalous conditions or events such as vehicle breakdowns, "control" communications interruptions, and malfunctions in guideway position-reference sub-systems. Though a difficult criterion to assess, it is perhaps one of the most important considerations in the design of an automated-vehicle system.

Consider for example one suggested procedure for handling the blockage of a guideway lane (resulting from, for example, a malfunctioning vehicle). Stefanek [37] considers the use of a bidirectional emergency center lane, onto which vehicles are re-routed, when a breakdown occurs on one of the regularly traveled lanes. Unfortunately, when a predisposition approach is used (Stefanek considers the "cycle" reservation type), this center lane would probably remain unallocated and hence "dormant" until needed to recover from an anomalous situation. Otherwise, all vehicles on the system, already scheduled on a path through the blocked segment, would have to be re-scheduled on a new path around the blocked lane. Such a re-scheduling operation would undoubtedly be a cumbersome task. (All affected vehicles could possibly exit from the system and then reenter it with a new schedule, however agrees limitations would possibly require that portions or all of the network be "slowed" down.)

In a deferred-disposition approach, however, a bidirectional center lane, such as the one just considered, could be utilized to
carry traffic in the direction of heaviest flow (during a given period). Were a breakdown to occur, the flow rate would be temporarily limited to a single-lane in each direction in the affected segment. Further only a local rescheduling of vehicles would be necessary because vehicles at distant points in the guideway network would not have, as yet, acquired a "rigid" schedule through the blocked area. Thus, deferred-disposition approaches would seem to be more fault tolerant than predisposition ones.

It would also appear that deferred-disposition algorithms would be better able to handle "exiting" backups which would occur (in dual-mode systems). These backups would occur because of

(1) Traffic jams on the conventional roadway or highway receiving the exiting vehicles.

(2) Limited storage and/or "processing" space for the exiting vehicles.

As exiting is a stochastic process with a random "service" time, predisposition macroscopic controllers are not able to preschedule this "event" and hence must rely on both exit storage areas and rescheduling procedures to handle exit "service" fluctuations.

Controller vulnerability was another aspect considered in the selection of a deferred-disposition approach. One feels that a system using a central master scheduler or controller is too susceptible to complete or partial shutdowns as a result of communications and/or central processing interrupts or failures. With deferred-disposition approaches, however, subcontrollers are utilized and need only be loosely coordinated by a central computer (for non-critical functions
such as routing-flow distribution).

The rationale for selecting both pre-interchange scheduling and dynamic maneuvering-space reservation are discussed in the next chapter.
CHAPTER III
THE DYNAMIC SCHEDULING APPROACH

In this chapter one approach to deferred-disposition macroscopic control is developed. The approach, termed dynamic scheduling, is described in terms of a generalized dual-mode network model. The controller architecture that is considered incorporates both a central coordinator and local interchange controllers.

A. A Generalized Dual-Mode Network

A representative portion of an automated dual-mode guideway system is depicted in Fig. 6. Note that this system interconnects with a conventional roadway system — the latter serving both as a collector-distributor and backup system to the former. The automated system is comprised of the following:

(1) Two-way guideway sections (single or multiple lanes in each direction),
(2) Interchanges where three or more guideway sections interconnect, and
(3) Entrance-exit facilities, to interface with the conventional roadway system.

In this chapter, it is assumed that the two-way guideway sections consist of a right-of-way with only a single automated lane in each direc-
Fig. 6—A representative automated guideway system.

B. Interchange Geometries and Flow Potential

Interchanges are areas where three or more guideway sections are interconnected. It is assumed that topography and available right-of-way would provide stringent controls on the selection of an inter-

Hence, only the longitudinal aspects of vehicle maneuvering need be considered. (The study of lateral maneuvering, i.e., lane-changing or weaving, is deferred until Chapter V where multiple-lane structures are considered.)
change for a particular site. These factors along with a joint consideration of both expected guideway flows and construction costs would probably be considered.

There are several basic interchange configurations for interconnecting guideways. In conventional highway systems a cloverleaf type of interchange is often employed. One such interchange, utilizing single-lane guideways, is shown in Fig. 7, from which it can be noted that only a single bridge is required. Loop ramps are employed to accommodate left-turning movements (1-B, 2-C, 3-D, and 4-A), and collector distributor lanes (a-n-s-j and h-r-o-c) are used to reduce the "interactions" between left-turning traffic and the through traffic in the east-west directions (1-C and 3-A). Note also, that "turn-around" movements are possible, e.g., a vehicle entering the interchange at Point 1 can traverse the interchange (via loops s-t and q-r) -- departing at Point A (on the same two-way guideway section in which it entered).

Two characteristics of such an interchange greatly constrain vehicle flows -- namely vehicle speed limitations on the loop ramps (due to their relatively small radii of curvature), and vehicle interactions as caused by the cloverleaf topology. The latter is most readily evident upon consideration of vehicle flows on the 12 normally traveled paths through the interchange ("turn-around" movements are not included here). Of these, however, there are six distinct arrangements of merges and diverges with representative ones being shown in Fig. 8. Some paths, such as 1-C, are clearly ones of minimum interaction be-
Fig. 7--A cloverleaf interchange configuration.
Fig. 8—Representative paths through a cloverleaf interchange.
cause "direct" ramps (here b-i) are provided, whereas other paths involve considerable interaction. Vehicles traversing 2-C, for example, "share" several guideway segments (interchange ramps) with other paths such as Segment p-m with Path 2-D and n-s with Path 1-B. Such sharing imposes stringent flow constraints. For example, assuming vehicles flow synchronously through interchanges\(^1\) (i.e., in moving slots as discussed in Chapter II), then for the path 2-C, the following constraints on the flow \(p_{2-C}\) are imposed\(^2\):

\[
\begin{align*}
\rho_{2-C} &\leq 1 - \rho_{2-A} - \rho_{2-D} \\
\rho_{2-C} &\leq 1 - \rho_{1-C} - \rho_{4-C} \\
\rho_{2-C} &\leq 1 - \rho_{2-D} - \rho_{3-D} \\
\rho_{2-C} &\leq 1 - \rho_{1-B}
\end{align*}
\]

where flow \(\rho_{i-j}\) is defined as the average fraction of slots at either point \(i\) or \(j\) that are occupied by vehicles traveling from \(i\) to \(j\). The first inequality, Eqn. (1), is based on "slot occupancy" of entering vehicles (Entrance-point 2), while the second on that of departing vehicles (Departure-point C). The third and fourth inequalities are based on the slot occupancies at merges p and n respectively. Were the lane b-i not present (i.e., a collector-distributor lane not in-

\(^1\)It is here assumed that due to limited space on the ramps within the interchange, all longitudinal maneuvering (to resolve merge conflicts) is accomplished outside the interchange proper.

\(^2\)It is assumed that there are no "turn-around" flows.
eluded), then Eqn. (4) would be replaced by the more stringent inequality
\[ \rho_{2-C} \leq 1 - \rho_{1-B} - \rho_{1-C} \] (5)

Constraints on entering or departing flows, similar to Eqns. (1) and (2), are applicable to all 4-way interchanges, regardless of the internal ramp configuration utilized, whereas additional constraints, such as Eqns. (3) and (4), are not always imposed. Consider, for example, the 4-way interchange configurations displayed in Figs. 9 and 10. Multiple-laned versions of these are utilized in conventional highway systems, and have been termed "all-directional" configurations as a "direct" ramp is provided for all directions of flow through the interchange. Since these ramps are not "shared," only entering and departing average flow constraints (Eqns. (1) and (2)) are imposed.

Although these directional interchanges have apparent flow advantages over the cloverleaf, they require additional costs. Only a single bridge is required for the latter, whereas six are required for the directional interchange shown in Fig. 9 and a four-level bridge structure for that in Fig. 10. Also note that "turn-around" movements are not permitted in these latter interchanges.

As one last point of comparison, it is interesting to consider the land requirements of each interchange type. If the interchange dimensions given in Figs. 7, 9, and 10 can be taken as "typical" in an automated highway context, the four-level directional interchange

3The dimensions of these drawings are based on those of similar conventional highway interchanges found in Reference 1. The lane widths have been shown larger than scale, however (for illustrative purposes).
Fig. 9—A 4-way directional interchange (multiple bridge structures).
Fig. 10—A 4-way directional interchange (four-level bridge structure).
would require less land than the multiple-structure one, which in turn would require less than the cloverleaf. In the latter, considerable land is needed to accommodate the loop ramps.

In the remainder of this study only directional interchanges are considered as it is felt that their high-flow rate capabilities are important to the efficient operation of a dual-mode automated vehicle system. However, this does not mean that cloverleaf type interchanges and/or partially-directional ones (those employing loops for some movements) would not be useful in an automated network. Such would certainly be useful where smaller flow rates are expected, as in rural areas.

Representative paths through these directional interchanges are displayed in Fig. 11. Note that for both interchanges the paths consist of either one or two diverges (where vehicles are "distributed" according to destination) followed by one or two merges (where vehicles are "collected"). Hence, both can be represented by the "distributor-collector" configurational model shown in Fig. 12 — the important attribute being that vehicle distribution precedes vehicle collection. (The manner in which this is accomplished is unimportant here.)

A 3-way model is easily obtained from this 4-way one by eliminating one distributor-collector pair (such as 2,B) and all internal links (ramps) serving same. Such a model is shown in Fig. 13, where it can be noted that only one-half as many links and considerably fewer bridges are required.
PATHS  4-way multiple structure interchange

1 --- a --- p --- D

1 --- a --- b --- g --- B

2 --- e --- f --- c --- A

1 --- a --- b --- f --- k --- C

PATHS  4-way four-level interchange

1 --- b --- k --- C

3 --- j --- i --- p --- o --- D

distribute  collect

Fig. 11—Representative paths through the directional interchanges.
Fig. 12—A distributor-collector interchange model.
Fig. 13—A 3-way and 4-way interchange, and a representative entrance-exit facility.
Also shown in Fig. 13 is an entrance-exit interchange facility. As was depicted in Fig. 6, such facilities would be utilized to interconnect a conventional highway system with an automated one. Such facilities are considered in more detail in Chapter V. However, it shall be noted here that it is desirable to provide storage for both entering and exiting vehicles, the first being useful in preventing short-term entrance demand peaks from necessitating vehicle rejection (and/or jams on the conventional highway) and the second for "absorbing" similar egress demands.

C. A Dynamic Scheduling Philosophy

In this section a dynamic scheduling approach for scheduling vehicles through an automated-vehicle system is presented. Other aspects of macroscopic control, such as routing procedures, entrance flow metering, and failure modes are not synthesized here.

Consider the representative portion of a guideway network shown in Fig. 14. Here a 4-way interchange (the shaded one) is displayed along with its "nearest neighbors" -- an entrance-exit facility, two 3-way, and a second 4-way interchange. These "neighbors" supply vehicles to, and accept vehicles from, the center 4-way interchange via two-way guideway sections.

1. Controller Architecture

The controller architecture for this representative subnet--
Fig. 14—A representative portion of a guideway network.

Fig. 15—The controller architecture.
work is shown in Fig. 15. Note that both the center interchange and each surrounding neighbor have separate, local controllers, which are coupled via direct communication links. Each local controller is also linked to a central coordinator.  

The important aspect of this architecture is the division of control tasks between the local and central levels of control. It seems desirable (from reliability considerations) to handle control tasks concerned with individual vehicles (e.g., scheduling and maneuvering) at the local level, and to assign those concerned with vehicles in the aggregate to the central level (e.g., "lane flow" distribution and "entrance rate" control). The assumptions implicit here are:

(1) Individual vehicle control tasks such as scheduling (e.g., through an interchange) and maneuvering are "critical" to safe, conflict-free operation;

(2) Communication reliability (for a fixed cost per unit distance) is inversely proportional to both communication link distance and transinformation rate (the product of information amount and its rate of transfer);

---

5In geographically extensive systems, the "coordinating functions" would be accomplished by a hierarchical system of subcoordinators, wherein area coordinators would "govern sectors containing several interchanges."
(3) Information processors (at both the local and central level) will be more reliable (for a given cost), if both the amount and rate of information processing required is small.

The realm of jurisdiction for the controller (of the center 4-way interchange) is depicted by bold lines in Fig. 16. Vehicle control is transferred from one interchange controller to the "next" immediately after a vehicle passes a "vehicle collection" point (merge area). Thus, lanes leading away from an interchange (the dashed ones for the center 4-way interchange) are within the governing area of the neighboring interchange controllers. Vehicles within an interchange proper (i.e., at one of the diverges, merges, or on one of the ramps) are, of course, within the realm of the corresponding interchange controller.

The guideway lanes approaching each interchange are included in the realm of interchange controller jurisdiction, as these lanes are needed for vehicle maneuvering space (necessary to avoid conflicts at the interchange merges). Here, it is assumed that there is insufficient distance on the ramps (internal links) of an interchange to accomplish a significant amount of vehicle maneuvering. The validity of this assumption is evident from a consideration of the interchange

---

6 Note that "entrance–exit facility" interchanges are included here, and distances between these would probably be less than 20 miles, even in rural areas.
Fig. 16—The realm of interchange controller jurisdiction.

dimensions (see Figs. 9 and 10).

2. The Interchange Controller

In Fig. 17 a situation has been depicted to illustrate the scheduling operations performed by an interchange controller. A vehicle traveling on the Route I-II-III (i.e., over the Path 1-2-3) leaves Interchange I at Point α where it comes into the jurisdiction of the controller associated with Interchange II (the controller of interest here). The following two functions are performed by the latter:
Fig. 17—Dynamic "path" scheduling.

(1) A **synchronous trajectory** is scheduled through the interchange over Path 2 so as to resolve any conflicts at the Merge-point $\gamma$. Further, a maneuvering trajectory is specified for the "interchange approach path" (Path 1), such that the vehicle arrives at Diverge-point $\beta$ at the required (scheduled) time.

(2) Also, to insure that the vehicle can be "handled" at the next interchange (III), **maneuvering space** must be
reserved on Path 3 (the approach to Interchange III).

The operational procedure by which the above scheduling functions would be performed is most easily understood after maneuvering trajectories have been considered in more detail.

3. Maneuvering Trajectories

A maneuvering trajectory is defined as the velocity-versus-position profile followed by a vehicle as it traverses a section of guideway. Consider a vehicle traveling over a fixed length "maneuvering section," such as the Path $\alpha - \beta$ in Fig. 17. If it were assumed that a vehicle would travel synchronously through each interchange, then the maneuvering trajectory would be constrained at the end points $\alpha$ and $\beta$. Thus, both vehicles entering the maneuvering section (at Point $\alpha$) and those departing it (at Point $\beta$) would do so at synchronous speed ($V_s$). However, between these points a vehicle could travel at an average speed either greater or smaller than $V_s$—depending upon whether it must be advanced (moved-up) or delayed (moved-back) with respect to the synchronous reference.

Several representative maneuvering trajectories are shown in Fig. 18. In the upper portion of this figure, velocity ($V$)-versus-position ($X$) profiles are plotted, while in the lower part, position ($X$)-versus-"time" ($t - X/V_s$) profiles are shown. This choice of time

---

7Positions 0 and 12 correspond to Points $\alpha$ and $\beta$, respectively, in Fig. 17.
Fig. 18—Representative maneuvering trajectories.
variable removes the profile "trend" that would otherwise be present (compare with the position-versus-time trajectories in Fig. 5). As a result the "position-time" profiles in Fig. 18 have zero slope in segments where vehicles are traveling synchronously. Such is the case over the whole maneuvering section for Vehicles A and C — both of which remain in "their" moving slot as they traverse the 12-slot distance.\(^8\) Also, for all other vehicles the "position-time" profiles have zero slope at the end points \((X = 0, X = 12)\) as operation is synchronous here.

Consider a move-up maneuver such as the one undertaken by Vehicle D. At position 0 this vehicle begins accelerating at a constant rate (hence, the slight parabolic curvature on the \(V\)-versus-\(X\) plot) until its speed is equal to some "maneuvering velocity," here \(4/3 \, V_s\) (at position 1). The vehicle then maintains this speed for an appropriate length of time before decelerating (at a constant rate) back down to synchronous speed. Note that the acceleration rates, the speed change, and the time durations were controlled so that a two-slot move-up (with respect to the synchronous reference) resulted. By an analogous procedure, Vehicle B was temporarily slowed down to accomplish a two-slot move-back.

The mechanics of such maneuvering procedures have been considered by Bender [4] who determined maneuvering distance requirements

\(^8\)The selection here of only 12 slots for the maneuvering section length was quite arbitrary (for simplicity in discussion) and in reality might be several orders of magnitude longer than this (e.g., 1200 one-hundred foot slots).
for an n-slot maneuver (move-up or move-back) as a function of allowable accelerations and maneuvering velocities. (The latter are constrained by passenger comfort and/or vehicle capability considerations.)

The important point to be noted here is that a vehicle stream, entering such a maneuvering section, may be "re-organized" — within certain limitations. Note in Fig. 18 how the Vehicles A, B, C, and D in the departing stream have been redistributed with respect to their relative positions in the entering stream.

The trajectories for Vehicles E, F, and G illustrate another useful maneuver. Here, vehicles are decelerated (in the examples here, at a constant rate) to a complete stop. After a vehicle has remained stationary for a given period of time it is accelerated (at approximately a constant rate) to bring it back up to synchronous speed. With such a maneuver, it is possible to delay a vehicle an indefinite number of slots. Note that at time 20 all three vehicles (E, F, and G) are stationary and "packed" at a separation of 1/5th of a slot. It is readily apparent that due to the finite length of the maneuvering section, the number of vehicles that can be so delayed is limited.

These trajectories illustrate one important capability required in any (realistic) automated vehicle system, that is, the operational ability to "store" vehicles within the guideway network. Not only would such "storage maneuvers" usually be necessary when a

---

9 Also, Godfrey [17] considers the mechanics of such maneuvers when the "rate of acceleration onset" is taken into account.
downstream blockage occurs (resulting from vehicle, guideway structure, and/or controller failures), but occasional backups at exit facilities could necessitate temporary storage of vehicles on the guideway.

Aside from anomalous situations, however, "storage maneuvers" are useful for handling situations where an interchange is subjected to heavy demand. In general, directional interchanges are flow limiting points (as will be shown in the next chapter), and the upper (average) flow bounds, per Eqns. (1) and (2), are usually unachievable. Hence, under heavy demand it may be necessary and/or desirable to store vehicles on the approach lanes to an interchange so as to both achieve high flow rates and to utilize guideway sections for some of the storage (otherwise accomplished at entrance storage areas). The questions of when and where to store vehicles are discussed in later chapters wherein several relevant factors including network topology and demand patterns are considered.

The storage capacity of a maneuvering section is a function of several factors including section length, maximum (allowed) decelerations, synchronous speed ($V_s$), and static vehicle packing density. Consider the state of a maneuvering section as depicted in Fig. 19. Here the section is in a state of maximum storage as vehicles along the length of this section are "packed" in stationary positions at some maximum allowed density (in this example 5 vehicles/slot length). Note however that on the ends of the section, space is required to decelerate from, or accelerate to, synchronous speed. The maximum allowed deceleration and acceleration (based on passenger comfort and/or
Fig. 19—A maneuvering section in the "state" of maximum vehicle storage.
vehicle-response considerations) would determine these speed-change profiles.

One final point to be considered is the assumption that there is, in general, insufficient space on the ramps of an interchange (as those depicted in Figs. 9 and 10) to accomplish all necessary maneuvering. For example, if available ramp distance were some 1000 ft, with time headways \( T_s \) of 1 sec, a synchronous speed \( V_s \) of 100 ft/sec, and a maximum acceleration (deceleration) rate \( a_o \) of 0.1 g (3.22 ft/sec\(^2\)), then the maximum number of slots that a vehicle could move-back would be only 1 slot.\(^{10}\) The profile for such a maneuver is of the form shown for Vehicle H in Fig. 18, i.e., a constant deceleration over one-half the available distance followed by an equal rate of acceleration over the remaining distance. One slot of maneuvering capability is clearly inadequate to resolve most merge conflicts -- even under moderate demand conditions. Even if twice the ramp distance (2000 ft) were available, no more than 5 slots of move-back could be accomplished on the ramp.

\(^{10}\) This estimate is based on an equation derived by Bender [4] for calculating the minimum distance required for an n-slot move-back maneuver:

\[
X_{\text{min}} = -n T_s V_s + 2 V_s \sqrt{n T_s V_s \over a_o}
\]
D. Interchange Clusters

The 3-way and 4-way interchanges shown in Fig. 13 can be "physically expanded" (so that there is considerable distance between the diverge-merge areas), and thus make it possible to achieve higher flow rates — as a result of a reduction in the interactions among maneuvered vehicles. The 3-way and 4-way "interchange clusters" shown in Fig. 20 result. Note that the "3-way Cluster" shown consists of three simple interchanges interconnected by guideway segments that can be utilized for vehicle maneuvering.

In the next chapter various models are constructed for both the 3-way and 4-way directional interchanges and the corresponding interchange clusters.
Fig. 20—Interchange "cluster" configurations.
CHAPTER IV
INTERCHANGE OPERATION

In this chapter both simple analytical queuing models and simulation models of interchange operation are developed. Analytical results obtained from the former are compared with empirical results obtained from the latter.

A. Analytical Queuing Models

Consider the two-way interchange shown in Fig. 2 and the corresponding queuing model as shown in Fig. 21. The dashed lines represent the approach lanes to the interchange, while the solid ones represent the ramps within the interchange proper. Each vehicle on an approach lane to the interchange is defined to be "in queue" for a period of time (measured in slot-times) equal to the number of slots of move-back required. (Move-up maneuvers are not considered here.) A vehicle proceeding at synchronous speed, without a move-back maneuver, is never "in queue" as no extra delay above synchronous travel time is incurred. This point was illustrated in Fig. 18, where, by an appropriate transformation on the time scale, synchronous travel time was "suppressed." Note that a vehicle entering an approach lane one slot behind another vehicle must be delayed at least as many slots as its
predecessor (e.g., see vehicles E, F, and G in Fig. 18).

In the model considered here, queue lengths (number of vehicles "in queue") were not limited and hence, the lengths of the approach lanes were not considered.

Returning to Fig. 21, note that the number of vehicles in Queue 1 and Queue 2 at $t = t_4$ are $N_1$ and $Q_1$ respectively. A vehicle is said to be "served" when it leaves the queue. The time between service in either queue is 1 slot-time or some multiple thereof. The service time for a vehicle at the front of a queue depends on its destination (here either A or B), the destination of the head vehicle in the other queue, and the priority structure. Because of merge con-
flicts, the service times for vehicles in either lane are strongly dependent on the arrival pattern in the other lane.

To analyze this queuing model in detail for a given demand distribution is a rather formidable task due to the interaction between service times and arrival patterns. Hence, only one statistic, maximum average per-lane flow, is determined here. (Other statistics as average delays are estimated in the next section by means of simulation models.) The following assumptions are made:

1. Order-of-arrival scheduling service is assumed. When two vehicles arrive at the same time, one of the two shall be randomly chosen to have priority.
2. Every arriving slot is occupied by a vehicle (since it is desired to determine the maximum obtainable flows).
3. The arrival distribution is stationary (time independent) and the arrival patterns in each lane are independent.

Consider the following origin-destination probability matrix:

\[
\begin{bmatrix}
\rho_{1A} & \rho_{1B} \\
\rho_{2A} & \rho_{2B}
\end{bmatrix}
= \begin{bmatrix}
\omega & \xi \\
\xi & \omega
\end{bmatrix}
\]

which was chosen to reduce the number of demand parameters\(^1\) from four

---

\(^1\) \(\rho_{1A}\) is the probability that in a given slot-time a vehicle arrives at Lane 1 with destination A. \(\rho_{1B}\), \(\rho_{2A}\), \(\rho_{2B}\) are similarly defined.
to two \((\omega, \xi)\). The following two demand parameters are defined:

A "demand level" parameter,

\[
\rho = \omega + \xi \tag{7}
\]

and a "demand balance" parameter,

\[
\gamma = \omega / \xi \tag{8}
\]

Per assumption (1)

\[
\rho = 1, \tag{9}
\]

and note that \(\gamma = 1\) for balanced demand (i.e., \(\rho_{1A} = \rho_{1B} = \rho_{2A} = \rho_{2B}\)). Due to symmetry in Eqn. (6) only the range \(0 \leq \gamma \leq 1\) need be considered.

There are four possible events, represented by arrival patterns for vehicles at the front of each queue. These are represented by the columns I, II, III, IV of the table shown in Fig. 21. The table entries (either letter A or B) are vehicle destinations and the row denotes the origin or queue (either 1 or 2). Events II and III result in no conflict as the "head of queue" vehicles in these cases proceed towards different merges; whereas, for Events I and IV only one of the two vehicles can be serviced in a given slot-time. The probability of each event is calculated from the demand matrix (Eqn. (6)) and heuristic values of same are simply the products of arrival probabilities (for each lane) as shown in Fig. 21.

Several simple observations are immediately apparent. First, if the "demand balance" parameter \(\gamma\) were equal to 0, then from Eqns. (7-9), \(\omega = 0\) and \(\xi = 1\), and from the event probabilities, Events I, II, and IV would never occur (prob = 0) and Event III would
occur every slot time (prob = 1).\(^2\) Since Event III involves no vehicle
conflict, every arriving vehicle would be "served" without delay.

Hence, the per-lane flow both into the interchange (Points 1 and 2) and
out of it (Points A and B) would be at capacity (1 vehicle/slot or
1 vehicle/slot-time). A similar demand situation, that also results in
no conflicts, occurs for \(\gamma = \infty\) (i.e., \(\omega = 1,\ \xi = 0\)), as here
all vehicles would proceed straight through the interchange.

A second observation is that the "probability of merge con­
flicts" (Events I and IV) is maximum (and it will be shown later that
output per-lane flow is minimized) for a balanced demand matrix, i.e.,
when \(\omega = \xi = 0.5\) (\(\gamma = 1\)). That is

\[
\text{prob (merge conflicts)} = \text{prob (I) + prob (IV)}
= \omega \xi + \xi \omega
= 2 \omega \xi
= 2 \omega (1 - \omega),
\]

which is maximum for \(\omega = 0.5\).

It would be useful to estimate per-lane maximum average flows
out of the interchange as a function of \(\gamma\). The output flow densities
are less than input flow densities for all \(\gamma\) (and \(\rho = 1\)) except the
special cases \(\gamma = 0\) and \(\gamma = \infty\), and it is easy to show that aver­
age flow is minimum for \(\gamma = 1\). Thus, the number of vehicles in each
queue continues to grow and delays approach infinity for all cases
except \(\gamma = 0\) and \(\gamma = \infty\). The output flow from the interchange can

\(^2\)The physical situation considered here when \(\gamma = 0\) is one
of no conflict since all vehicles would make a turning movement (confer
Fig. 2).
be heuristically obtained by using a simple conservation-of-vehicles relationship:

\[
\begin{align*}
\text{flow rate out} & = \text{flow rate in} - \text{Sum of average queue growth rates} \\
& = 2 - \left[ 1 \cdot \text{prob (I)} + 1 \cdot \text{prob (IV)} \right] \\
& = 2 - 2 \omega \xi
\end{align*}
\]

and per-lane flow (\( \rho_0 \)) is simply half of this:

\[
\rho_0 = 1 - \omega \xi
\]

The queue growth rates were obtained via the following reasoning:

1. In a given slot-time, if either Event II or III occur, then two vehicles are served, and two arrive (in separate lanes of course), and thus the number of vehicles "in queue" does not change, i.e.,

\[ N_{i+1} = N_i \text{ and } Q_{i+1} = Q_i. \]

2. If, however, either Event I or IV occur, then (due to a merge conflict) only one vehicle can be served in that slot-time. Since two vehicles always arrive to add to each queue, then one of the two queues (based on a random selection) is increased in number by one, i.e., either
\[ N_{i+1} = N_i + 1 \text{ and } Q_{i+1} = Q_i \],

or

\[ N_{i+1} = N_i \quad \text{and} \quad Q_{i+1} = Q_i + 1 \].

Unfortunately, Eqn. (11) cannot be expressed directly in terms of \( \gamma \) alone. However, by calculating \( \omega \) and \( \xi \) for a given \( \gamma \) (using Eqns. (7-9)), it is apparent that \( \rho_0 \) decreases monotonically from 1 for \( \gamma = 0 \) to 0.75 for \( \gamma = 1 \). The important result here is that maximum average per-lane flow out of the interchange is reduced to only 75% of guideway capacity as demand approaches the balanced state.

If a 3-way interchange (as in Fig. 13) were considered and a similar queuing model applied (see Fig. 22), then for the demand matrix:

\[
\begin{bmatrix}
0 & \rho_{1B} & \rho_{1C} \\
\rho_{2A} & 0 & \rho_{2C} \\
\rho_{3A} & \rho_{3B} & 0
\end{bmatrix}
\begin{bmatrix}
0 & \omega & \xi \\
\xi & 0 & \omega \\
\omega & \xi & 0
\end{bmatrix}
= (12)
\]

and for the same assumptions as with the two-way model:

Total average flow out = 3 - 1 \cdot [ \text{prob (I)} + \text{prob (II)} + \text{prob (IV)} + \text{prob (V)} + \text{prob (VII)} + \text{prob (VIII)} ]
Fig. 22—A 3-way interchange queuing model.

\[ 3 - \left[ \omega^2 \xi + \omega \xi^2 + \omega^2 \xi \right] \]
\[ + \xi^2 \omega + \omega^2 \xi + \xi^2 \omega \]
\[ = 3 - [3 \omega^2 \xi + 3 \xi^2 \omega] \]

The per-lane flow is simply one-third of the total flow:

\[ \rho_o = 1 - \omega^2 \xi - \xi^2 \omega \]

\[ = 1 - \omega \xi \] (13)
Eqns. (7), (8), and (9) were used in the latter simplification. This result is identical to that obtained for the 2-way interchange (Eqn. (11)).

As would be expected, the same result (Eqn. (13)) is also obtained for a 4-way interchange (see Fig. 13) subject to the demand:

\[
\begin{bmatrix}
0 & \rho_{1B} & \rho_{1C} & \rho_{1D} \\
\rho_{2A} & 0 & \rho_{2C} & \rho_{2D} \\
\rho_{3A} & \rho_{3B} & 0 & \rho_{3D} \\
\rho_{4A} & \rho_{4B} & \rho_{4C} & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & \omega & 0 & \xi \\
\omega & 0 & \xi & 0 \\
0 & \xi & 0 & \omega \\
\xi & 0 & \omega & 0
\end{bmatrix}
\quad (14)
\]

In Fig. 23 Eqn. (13) has been plotted as a function of demand balance (\( \gamma \)). Empirical results obtained by simulation models (discussed in the next section) are also shown.

B. Simulation Models

In this section the operation of a "maneuvering section" or approach lane to an interchange is discussed. Interchange simulation models are then developed and empirical results from same are presented.

1. Maneuvering Section Operation

In the previous chapter, the maneuvering of vehicles on the approach lanes to an interchange were discussed in some detail. The purpose of such maneuvering was to resolve interchange conflicts so that vehicles could be safely and smoothly sequenced through the merges.
Fig. 23—Maximum average per-lane flow density versus demand balance ($\gamma$).

\begin{align*}
\text{DEMAND BALANCE (\(\gamma = \omega / \xi\))} \\
\text{FLOW DENSITY (veh/slot)}
\end{align*}
within an interchange. Also, it was assumed that all delays would occur externally to the interchange and that operation within same would be synchronous.

The scheduling and maneuvering of vehicles would be under the jurisdiction of the local controller associated with an approached interchange. This controller must be capable of "overseeing" the following modes of operation:

(a) Normal scheduling;
(b) Deferred scheduling; and
(c) Anomalous operation.

In the "normal scheduling" mode a path is scheduled for a vehicle (through the approached interchange) when it arrives at the beginning of the maneuvering section, i.e., at Point $\alpha$ in Fig. 17. Thus, the time at which the vehicle is to depart the section (at Point $\beta$) is specified by the scheduling strategy. From this the required number of slots of move-back is calculated and an appropriate move-back maneuver determined (such as the one undertaken by Vehicle B in Fig. 18). In addition, maneuvering space is reserved for the approach to the "next" interchange. If the latter cannot be accomplished, then a "deferred-scheduling" mode of operation is entered. Here, the task of scheduling the vehicle through the "approached" interchange is deferred indefinitely by causing the vehicle to follow a gradual deceleration trajectory, which, if it were followed for a sufficient length of time, would bring the vehicle to a complete stop (i.e., in a

---

3Move-up maneuvers are not considered here.
stationary state along with other such vehicles). Such trajectories are illustrated by vehicles E, F, and G in Fig. 18. If, however, during the deceleration maneuver, the local controller were able to reserve maneuvering space in the approach to the "next" interchange, a path would immediately be scheduled through the "approached" interchange and an appropriate acceleration profile would be followed to meet same. (Note the paths for vehicles H and I). If, however, vehicles must be brought to a complete stop (vehicles E, F, G), they would be delayed until they could be both scheduled through the interchange and reserve maneuvering space. Such vehicle stoppage might occur as a result of unsatisfactory operation of the central coordinator -- which controls the demand on each interchange by both judicious routing pattern control and entrance flow regulation. (However, it may even be found that stopping vehicles on guideway sections is desirable to reduce entrance rejection rates and/or entrance storage requirements.) Besides "normal" and "deferred" scheduling modes, there would of necessity be an "anomalous" mode to handle breakdowns.

2. Interchange Models

The rationale behind the choice of computer simulation models to study interchange operation (rather than comprehensive analytical analysis) includes the following:

(1) Simulation models are quite amenable to "block-building" techniques and hence the study of larger networks composed of several interconnected
interchanges.

(2) Although simulations do not result in closed-form solutions as is sometimes possible with analytical analysis, the accuracy of statistical estimates, obtained experimentally via simulation, is generally adequate. Further, the many gross assumptions, often necessary in an analytical approach (to avoid complicating an overall model), are usually not necessary.

(3) Due to both the flexibility and experimental nature of the simulation technique, the researcher has considerable freedom in "dynamically" choosing the "direction" of the study and the hypotheses to be tested.

(4) Finally, a simulation model is quite useful as a "thinking" tool for both the researcher and others with whom he communicates. (In an analytical model, simple concepts are often obscured by cumbersome mathematics.)
In the following subsections (2.1, 2.2, 2.3), several basic simulation "elements" or model building blocks are defined and used to construct interchange models. The details concerning the computer realizations of these models are presented in the Appendices together with the operation and use of the Deferred-Disposition Transport Simulator (DDTS), which is the software system developed for this study. It is a general-purpose system which can be easily used to simulate a large variety of both single-laned and multiple-laned guideway networks.

2.1 Basic Model Elements

Typically, a simulation model is constructed by interconnecting many small "building blocks" or model elements. The operation of each element is usually quite easily understood; however, the operation of the complete model can be quite complex due to the many interactions among the interconnected elements.

The model elements of interest here are shown in Fig. 24. To construct a simulation model these elements are interconnected to form the required configuration of guideways. The "entities" flowing through these elements are vehicles, and for the symbols shown, flow is from left to right. There are also information flows interconnecting these elements — thus, simulating communication and control aspects.

Each element has at least one input node and one output node. Simulation "events" are defined as the arrival or departure of vehicle entities at these nodes. Since the elements have been chosen such that synchronous movement occurs at the element node points, events only
### BASIC MODEL ELEMENTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Mnemonic</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinK</td>
<td>LK</td>
<td><img src="image" alt="LinK Symbol" /></td>
</tr>
<tr>
<td>Store/Gate</td>
<td>SG</td>
<td><img src="image" alt="Store/Gate Symbol" /></td>
</tr>
<tr>
<td>Organize</td>
<td>OZ</td>
<td><img src="image" alt="Organize Symbol" /></td>
</tr>
<tr>
<td>Source/Sink</td>
<td>SS</td>
<td><img src="image" alt="Source/Sink Symbol" /></td>
</tr>
<tr>
<td>MerGe</td>
<td>MG</td>
<td><img src="image" alt="MerGe Symbol" /></td>
</tr>
<tr>
<td>DiverGe</td>
<td>DG</td>
<td><img src="image" alt="DiverGe Symbol" /></td>
</tr>
</tbody>
</table>

*Fig. 24—Basic simulation-model elements.*
occur at discrete time intervals corresponding to slot-times. Thus, the time it takes for a vehicle (entity) to "travel through" an element (i.e., the time between its arrival and departure events for a particular element) is an integral multiple of slot-times.

The operation of each of the six elements are as follows:

(1) The Link element (LK) represents a section of synchronous guideway. Hence the travel time through such an element is equal to a constant representing the length of the link. A link thus represents a fixed delay to a vehicle.

(2) The Store/Gate element (SG) delays vehicle entities by a variable amount. This element can be set (by an information input) to one of two states: "gate open" or "gate closed." In the "gate-closed" state no entities are allowed to leave the element and hence, arriving ones are accumulated in a storage queue. Once the gate is opened, accumulated entities depart (in the same order in which they arrived) at the rate of one per slot-time. If a vehicle entity arrives when the gate is open and when no entities are currently in the
"storage," it traverses the (SG) element in a single slot-time. A SG element has a capacity, which represents the maximum number of entities it can simultaneously store. SG elements are used to simulate the operation of "deferred scheduling" (as is later discussed).

(3) The Organize element (OZ) is similar to an SG element in that it represents a variable delay to entering vehicle entities. Two or more OZ elements are logically "grouped" together (in a control sense) via a scheduler module, which ensures that two or more entities with common destinations do not depart simultaneously from OZ elements in the same group. A 2-way interchange model uses two such elements, whereas, 3-way and 4-way ones use groups of three and four elements, respectively. The minimum travel time through an OZ element is equal to an attribute termed its length. The scheduler can delay vehicle entities by an amount additional to synchronous
travel delay to ensure that no more
than one entity with a given destina-
tion departs the group at the same
time. Entities leave an element in
the same order in which they arrive.

(4) The Source/Sink element (SS) is the only
simulation element where vehicle enti-
ties can be created or destroyed. The
source portion of this element (output
node) creates entities (and specifies
their destinations) according to an
origin/destination demand matrix (which
can be time-dependent). Special enti-
ties representing empty slots are also
created when necessary. The creation
of vehicle entities are accomplished
by Bernoulli trials (using pseudo-
random numbers).

The sink portion of the SS element
(input node) is used to represent ve-
hicles leaving the "system." Besides
destroying vehicle entities, this element

4Vehicles may also be destroyed when an SG or OZ element ex-
ceeds capacity or when two vehicles "collide" at a merge element; how-
ever, these are treated as simulation error conditions.
has the task of collecting travel-time data so that delay statistics can be calculated.

(5) The **Merge** element (MG) combines two vehicle flows. Travel time through the merge is always one slot-time. If vehicles arrive simultaneously, at the two input nodes to a merge, a "collision" (and hence simulation error) is considered to have occurred.

(6) The **Diverge** element (DG) segregates vehicles according to their destination. A "routing" table, associated with each DG element, determines which output node is selected. As with the MG element, the travel time through the DG element is one slot-time.

2.2 A Two-Way Interchange Model

A 2-way interchange (α), along with two "downstream" neighboring interchanges (β and γ) are shown in Fig. 25. The latter are included so that the inter-interchange information flow can be defined. Also shown in this figure are the simulation element connections for modeling the operation of one of these interchanges. Note that the interchange proper is modeled by two diverge (DG), two merge (MG) and
Fig. 25—A 2-way interchange and its downstream neighbors. A 2-way simulation model is also shown.
four link (LK) elements. The lengths of each of the latter are assumed to be the same. Hence, the travel time through the interchange on any of the four possible paths (DG - LK - MG) is the same.

Note that in each approach lane, an Organize element (OZ) supplies vehicles to the interchange. The two such elements required here are controlled by a scheduler so that in a given slot-time both do not "output" a vehicle going to the same destination (either A or B) and hence to the same merge point. Thus, the OZ elements either output vehicles going to different merges (one to A and one to B), or one OZ outputs a vehicle while the other outputs an empty slot (entity). Since all four (DG - LK - MG) paths are of equal length, this organizing procedure resolves all merge conflicts.

The scheduling strategy utilized is order-of-arrival scheduling of vehicle entities (as they arrive at the input nodes of the OZ elements). When two arrive simultaneously, the one in Lane 1 is scheduled prior to obtaining a schedule for the one in Lane 2. (However, both scheduling operations (decisions) are performed within the same slot-time.)

A SG element is also incorporated into each approach lane so as to allow the scheduling operation to be deferred (as previously discussed). Normally vehicle entities pass through this element with only one slot-time of delay. However, when necessary, the gate at the out-

---

5 The length is rather unimportant to the simulation as these elements only result in a small fixed delay -- compared to the long variable delay between interchanges.
put node of the SG element can be closed — allowing entities to accumulate within the element (up to its capacity) and hence be delayed indefinitely.\(^6\) An SG gate is closed under either of the following two conditions:

1. The immediately following OZ element has reached capacity and/or,
2. The number of vehicles in either of the SG elements on the approach paths to one of the downstream interchanges (\(\beta\) and \(\gamma\)) is greater than some given fraction of capacity (a threshold parameter is involved here).

Every slot time the above two conditions are tested, and the appropriate gate "state" is set. The testing of the second condition above, represents the task of reserving maneuvering space in the approach to the "next" interchange. If space cannot be reserved (i.e., the SG of the downstream interchange currently contains more than a given number of vehicles), the scheduling of the vehicle is deferred. (This is accomplished in the model by closing the gate of the SG in the currently approached interchange.)

The required "flow of information" between model elements is

---

\(^6\)The SG elements used here do not represent "physical vehicle storage areas," but rather provide "lumped" representations of the delay associated with deferred-scheduling maneuvers (see Figs. 18 and 19). Each (SG, OZ) pair provides the total required delay associated with maneuvering-section travel.
illustrated in Fig. 26 where elements for the three interchanges \((\alpha, \beta, \gamma)\) are shown; however, only the information-flow paths for Interchange \(\alpha\) are explicitly shown. The orientation of elements has been displayed such that the direction of all vehicle flows are from left to right; thus, the elements for the two downstream interchanges \((\beta, \gamma)\) are on the right. Note that the direction of information flow (right to left) is opposite to that for vehicle flow (left to right). Thus from a control system aspect, vehicle entity flows provide "feedforward" paths, whereas, information flows provide loop-closures or "feedback" paths.

The four representative information paths (I, II, III, and IV) provide the required inter-element information flows for the vehicle approach Path 1. Information Path I is a bi-directional link between the "parallel" OZ elements (via the interchange scheduler). It is required as the operation of these two elements must be carefully coordinated. The feedback necessary to close the SG gate when the OZ element reaches capacity is provided by Path II. This gate can also be closed via information from Paths III and IV (thus deferring scheduling of vehicles) if the number of vehicles in either of the SG elements leading to Interchanges \(\gamma\) and \(\beta\), respectively, is greater than some given threshold value.

2.3 Other Interchange Models

Models for both the 3-way and the 4-way interchange (see Fig. 13) are quite similar, in many respects, to the 2-way one. The approach lanes in each case are modeled by (SG, OZ) element pairs. The
Fig. 26—The required "information flow" paths.
3-way model, shown in Fig. 27, requires three such pairs; whereas, the 4-way one, shown in Fig. 28, utilizes four. Each OZ element associated with a given interchange is tied in with a "scheduler module." The strategy to be considered is order-of-arrival\(^7\) and simultaneous arrivals are handled by a fixed priority structure as before (the order is unimportant here).

For proper sequencing through the merges, the travel time for each path through the interchange (from an OZ element output node to a departing merge output node) must be the same.\(^8\) Thus in the 3-way model each LK element is of the same length, whereas, in the 4-way one, some elements are shorter by two slots due to the diverge and merge pairs required here. (In the latter, if special 3-lane merge and diverge elements were used, all LK elements would be of the same length.)

As with the 2-way interchange, information feedback paths are required from each OZ element to its adjacent SG one. When two or more interchanges are interconnected, an information network linking same is required as was the case with the three 2-way interchanges already considered. More will be said concerning information flows in later chapters when specific interchange networks are considered.

---

\(^7\)Sluss [36], in a sub-study to this one, considers several different disciplines for scheduling vehicles through a 3-way interchange. Only an order-of-arrival strategy is considered here (and in the various interchange networks considered in later chapters).

\(^8\)In a real-world system, the interchange schedulers would compensate for differences in path lengths (i.e., various length ramps).
Fig. 27—A 3-way interchange simulation model.
Fig. 28—A 4-way interchange simulation model.
3. Experimental Interchange Studies

To evaluate interchange (model) performance, source streams must be provided for each approach lane and statistics gathered concerning the flow of vehicles through the model. One method for accomplishing this is to connect Source-Sink (SS) elements to the appropriate input and output nodes of the model. In Fig. 29 a single such element is connected to one pair of input-output nodes (3, C) of a 3-way model to form an "entrance-exit interchange-facility." By similarly connecting SS elements to the other "open" node pairs (here, 1, A and 2, B), one has an operational model of an isolated 3-way interchange.

The choice of model parameters for the OZ and LK elements of this isolated network model is relatively unimportant as the choice of length parameters for these elements has no effect on the measured statistics, and further the capacity parameters (for these elements) are only important when large delays are encountered (as a result of heavy demands). OZ and SG capacities of 40 and 200 slots respectively were used. If the number of vehicles in an SG element were to reach its capacity, the vehicles from the preceding SS element would be rejected. (In the simulation — entities are destroyed.) This would generally represent an excessive demand situation where queues would grow indefinitely if allowed and delays would grow without bound.9

9SG element over-capacity situations never occur when the interchange is in an interchange network environment and proper information feedback is employed. This is demonstrated in later chapters.
Fig. 29—An entrance-exit interchange-facility simulation model.
A series of experiments were conducted (using the DDTS software system) to determine both average per-lane flow rates and vehicle delays for this interchange. The demand matrix employed was that of Eqn. (12) with various combinations of "demand level" and "demand balance." The SS elements generate entities and their destinations via independent Bernoulli trials. All experimental runs were initialized with an empty interchange (i.e., with no vehicles in any of the elements), and were run for 5000 slot-times. Data were collected over this time and no attempt was made to remove biases resulting from initial transients occurring before a steady state was reached.\(^{10}\)

Two statistics were calculated for each simulation run:

1. The average per-lane vehicle flow density -- obtained by observing slot-occupancy at the departure points of the interchange (A, B, and C in Fig. 27);

2. The average delay -- obtained from data gathered as each vehicle arrived at a sink node. Here, measured source-to-sink travel times (minus the source-to-sink distance in slots) were used.

\(^{10}\)This is justified by the small size of the network (short travel times), and further, only a relative comparison of vehicle flows and delays for various demand parameters was of interest.
Experimentally obtained flow density characteristics are shown in Fig. 30. Here the average interchange per-lane output flow density is plotted as a function of the entering-vehicle "demand level" for various values of the "demand-balance" parameter. The flow-limiting nature of the interchange is clearly shown by these curves as the output flow equals input flow (demand level) for all demands below a certain "saturation level" — which is dependent on $\gamma$. Since $\gamma = 0$ represents a no-interaction case (here, at each merge one stream contains no vehicles), "saturation" did not occur and hence output flows reached 1 vehicle/slot-time. However, as $\gamma$ was increased to unity (balanced demand), the saturation level was depressed to 0.75. This is also illustrated in Fig. 23 where both theoretical and experimental results are shown.\footnote{Experimental results are also shown here for the 2-way and 4-way interchanges subjected to the appropriate demand conditions.}

Experimental results obtained for the 4-way model subjected to a balanced demand (all entries in the demand matrix are the same, except those on the diagonal are zero) also indicate a saturation flow level of 0.75.

The delay characteristics for the 3-way interchange are shown in Fig. 31. In Fig. 32 the same data are plotted using a log scale. In these figures average delays are plotted as a function of $\rho$ with $\gamma$ as a parameter. Note that delays increase rapidly as the saturation demand level (for the appropriate $\gamma$) is approached. As expected, there are no delays for $\gamma = 0$ as no interactions occur.
Fig. 30—Experimentally obtained flow-density characteristics for a 3-way interchange.
Fig. 31—Experimentally obtained delay characteristics for a 3-way interchange.
Fig. 32—Experimentally obtained delay characteristics for a 3-way interchange (log scale).
C. Interchange Clusters

The 3-way and 4-way interchange "cluster" configurations shown in Fig. 20 are simply special directional configurations allowing considerable distance prior to each merge area for longitudinal maneuvering. Also, maneuvering can be accomplished on the approach lanes to the cluster.

It is quite clear that maximum per-lane flow densities, out of the cluster, approach 100% as pre-merge maneuvering distances are increased (so that no maneuvering is necessary on the approach lanes to the cluster). However, delays would approach infinity as output flows approached 1 vehicle/slot-time (except for special cases as $\gamma = 0$ or $\gamma = \infty$) because expected maneuvering delays for a simple merge tend to infinity when the sum of the two input flows is increased towards unity [17].

1. Cluster Models

Simulation models were constructed to determine the pre-merge maneuvering requirements for achieving high per-lane output flow rates (e.g., .90 - .98). A 3-way interchange-cluster model is shown in Fig. 33 and a 4-way model in Fig. 34. The basic DDTS model elements (see Fig. 24) are utilized in these models, with both the pre-merge and the pre-cluster maneuvering areas being modeled by (OZ, SG) pairs, and as before, feedback is used from each OZ to its preceding SG element. As only a limited amount of maneuvering can be accomplished on the internal (pre-merge) sections, feedback is required between these and the "cluster-approach" maneuvering sections. Thus, for example, in the
Cluster-3 model

Fig. 33—A 3-way interchange-cluster model.
Fig. 34—A 4-way interchange-cluster model.
model shown in Fig. 34, when the number of vehicles in either SG elements 6, 14 and/or 11 is greater than some threshold value, the gate for SG element 3 (in approach lane 3) is closed — resulting in a temporary deferring of the scheduling operation for vehicles entering that approach section.

2. Cluster Model Experiments

Both the 3-way and 4-way interchange-cluster models were simulated for 10,000 slot-times under balanced demand (with 1 vehicle/slot-time at each approach), and with the following element parameter values:

- LK Length = 5 slots;
- OZ Length = 50 slots;
- OZ Capacity = 100 vehicles;
- SG Threshold = 100 vehicles; and
- SG Capacity = 200 vehicles.

Per-lane average flow rates out of the 3-way and 4-way cluster were .964 and .970 vehicles/slot-time respectively. These results clearly indicate an improvement over the .750 flows obtained for similar demand when pre-merge maneuvering was not possible.

To access the affect of reducing the available maneuvering space, the 3-way cluster was tested with the following parameter values:

- LK Length = 5
- OZ Length = 10
- OZ Capacity = 20
- SG Threshold = 20
SG Capacity = 40.
The per-lane average flow rate for this case was .90. In this latter case less pre-merge maneuvering was possible, and hence, flows were reduced by interactions among vehicles maneuvered on the approach lanes to the cluster.
A corridor network is defined as a series of entrance-exit facilities linked by two-way guideway sections. Such a network differs from a "line-haul" mass-transit system in that for the latter, auxiliary "feeder" systems such as bus lines are often necessary; whereas, in the former, the collector-distributor function is provided by the manual operation of dual-mode vehicles on conventional roadways and streets.

The following important aspects of corridor network operation are studied here:

1. Entrance-exit interchange (configuration) selection,
2. Lane-changing on multiple-laned guideway segments, and
3. Entrance rejection under heavy demands.

A. Alternative Configurations

The three representative corridor networks — LINE 1, LINE 2, and LINE 3 — shown in Fig. 35, each have four entrance-exit areas (I, II, III, and IV). The performance characteristics of these networks
Fig. 35—Three corridor networks.
are probably similar to those of larger networks (with more such areas); thus, only the former were considered.

The end facilities (I and IV) of each configuration are simpler than the center ones (II and III) as interchanges are not needed here. However, the center facilities are representative of both those along a long corridor and those in a loop network.

1. LINE 1 Network

In the LINE 1 Configuration, 3-way interchanges are utilized at entrance-exit areas, and only a single guideway lane is provided in each direction as shown in Fig. 36. Note that both entrance and exit lanes at each facility incorporate vehicle storage areas where the service priority is assumed to be on a first-in-first-out (FIFO) basis. When manually controlled vehicles arrive at an entrance they would either be allowed to enter the system or (if the storage were full) they would be rejected.¹ (Other aspects of inputting a vehicle to the system such as manual-automatic mode switching [35], vehicle inspection, and destination selection by the user are not considered here.)

The shaded segments of guideway (in Fig. 36) represent areas where longitudinal maneuvering of vehicles is permissible; whereas, non-shaded areas are those of synchronous vehicle movement. First, note that maneuvering is permitted in each approach to an interchange so as to resolve merge conflicts. Second, a vehicle entering the sys-

¹In a real-world system potential automated highway users would probably receive "radio forecasts" concerning the status of the system and hence entrance rejection need not be unanticipated.
Fig. 36—The LINE 1 Configuration.
tem must be accelerated to synchronous speed on a ramp in the entrance storage area and meet a scheduled slot position (for travel through the interchange). (A closed-loop "slot tracking procedure" similar to that suggested by Eliassi [14] for ramp-vehicle merging could be used for such maneuvering.) Also, vehicles leaving the system through an interchange must be decelerated from synchronous speed (while in the exit storage area) if automatic-to-manual mode switching were to occur at a slow speed.

One readily apparent shortcoming of the LINE 1 Configuration is the strong interaction between east- and west-bound traffic. For example, if there were a heavy demand for travel from Facility IV to III, then vehicles entering at II and traveling to I, could be delayed by preceding vehicles (in the same storage) with a destination at Facility III. This would result because of delay to eastbound vehicles waiting at Facility II — to obtain maneuvering space reservations for the eastbound approach to III.

2. LINE 2 Network

The LINE 2 configuration is comprised of two distinct guide-ways — one each for eastbound and westbound traffic — with separate entrance-exit facilities as shown in Fig. 37. Unfortunately, the use of separate entrances does not completely solve the problem of interaction between east- and west-bound traffic — it merely shifts the interaction point back to the conventional highway system where entering vehicles are segregated with respect to direction (here, either east or west). Thus consider the alternate layout for this facility shown in
Fig. 37—The LINE 2 Configuration.
Fig. 38. If the merge and diverge areas of the conventional highway were included as part of the facility, then the paths followed by entering and exiting vehicles would be similar to those for the 3-way interchange facility (used in the LINE 1 Configuration). However, an important difference is that in the LINE 2 Configuration the entrance and exit storage facilities are between diverge and merge areas and hence pre-merge maneuvering of vehicles would be effectively achieved. Per the discussion in Chapter IV, this should result in fewer interactions and hence higher obtainable flow rates.

3. LINE 3 Network

In the LINE 3 Configuration, two lanes are available for each direction of travel, and separate entrance-exit facilities are provided for each direction as shown in Fig. 39. At least two major system improvements are expected with the use of both this and other multiple-lane configurations:

1. The level of service would be increased as a result of higher flow capacities and smaller travel delays;

2. Alternate paths would be available should a single lane become blocked — due to anomalous conditions such as vehicle failures or lane shutdowns (e.g., for guideway structure maintenance).
Fig. 38—The parallel entrance-exit facility.
Fig. 39—The LINE 3 Configuration.
Both macroscopic and microscopic control operations for a multiple-laned network are more complex (than for a corresponding single-laned one) as both lateral (lane-changing) and longitudinal maneuvers could be utilized. In an automated highway system where vehicle lateral positioning would be accomplished by electronic steering controllers [15], [33], rather than mechanical guideway steering [31] (e.g., monorail systems), lane-changing maneuvers would be feasible at any point along the right-of-way. However, in practice such maneuvers need not be possible at every point along the guideway.

In the LINE 3 Configuration, shown in Fig. 39, special sections, here termed "swerve areas," are utilized for lane changing. The policy considered in this study is a rather simple one and is only intended to provide insights into lane-changing procedures rather than an exhaustive study of same. Vehicles in the two approach lanes to a swerve area would undergo longitudinal maneuvering to resolve potential conflicts between vehicles in the swerve area. The following assumptions are made:

1. Vehicles would travel at synchronous speed through the swerve area;
2. The moving slots in the two lanes would be in synchronism;
3. The swerve area would be of sufficient length to allow a vehicle to change lanes by a simple lateral maneuver (passenger comfort and/or vehicle...
capabilities would be factors to consider in selecting this length).

The longitudinal and lateral maneuvering areas for a two-lane situation are illustrated in Fig. 40. The operation would be as follows: When a vehicle arrives at Point $\alpha$ in either Lane 1 or Lane 2, the lane that the vehicle is to occupy when it departs the lateral maneuvering area at Point $\gamma$, would be selected. Should the vehicle be required to undergo a lane-changing maneuver, the latter would be initiated at $\beta$ and completed at $\gamma$.

![Diagram of longitudinal and lateral maneuvering areas](image)

**Fig. 40**—The longitudinal and lateral maneuvering areas necessary for lane-changing operations.

---

^2^This decision would probably be made by a local controller (governing lane-changing operations. The central coordinator would specify the routing patterns to be followed. For example, the proportion of vehicles (with a given destination) in each lane would be dictated by the central coordinator; whereas, the lane designation for a particular vehicle would be determined by the local controller.
There are nine possible "events" or combinations of "lane-changing designations" at Point α — as depicted in Fig. 41. The first six (I–VI) clearly represent acceptable combinations at the downstream Point β as none would result in a vehicle conflict. Event VII represents two side-by-side vehicles which must "swap" lanes, while Events VIII and IX represent adjacent vehicles, both of which are to be in the same lane when Point γ is reached. Longitudinal maneuvering must be utilized to insure that the latter two events do not occur at Point β as this would result in two vehicles occupying the same slot at Point γ. Event VII, however, is here considered as an acceptable condition at Point β as the two vehicles involved would occupy dif-

\[\begin{array}{cccccccc}
1 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
\text{Lane 1} & \bullet & \rightarrow & \bullet & \rightarrow & \nearrow & \bullet & \rightarrow & \nearrow \\
\text{Lane 2} & \bullet & \bullet & \rightarrow & \rightarrow & \bullet & \nearrow & \nearrow & \rightarrow \\
\end{array}\]

\[\begin{align*}
\text{→} & \text{ No Lane change} \\
\nearrow & \text{ Lane 1 to Lane 2} \\
\rightarrow & \text{ Lane 2 to Lane 1} \\
\bullet & \text{ Empty Slot}
\end{align*}\]

Fig. 41—Possible combinations of lane-changing requests for two adjacent lanes.
ferent slots at departure Point $\gamma$. To prevent a collision in the lateral maneuvering area, as the vehicles "swapped" lanes, one vehicle would have to be in the front portion of its slot and the other one to the rear.\(^3\)

It is clearly apparent that vehicles entering at Point $\alpha$ must be scheduled through the lateral maneuvering area ($\beta - \gamma$) so that conflicts (Events VIII and IX) do not occur as lane-changing maneuvers take place. An interesting observation is that both the scheduling and the longitudinal maneuvering operations are conceptually identical to those required for a 2-way interchange per Fig. 42.

B. Corridor Models

In this section, simulation models are developed for the three corridor configurations, and in the next section, experimental comparisons of guideway flow rates, vehicle travel delays, and entrance rejection rates are presented.

1. LINE 1 Model

A simulation model for the LINE 1 Configuration is shown in Fig. 43. The two center entrance-exit interchange facilities (II and III) are modeled by interconnecting two of the "E-E Models" shown in Fig. 29. Note that the storage and scheduling of entering vehicle entities is accomplished by (SG,0Z) element pairs at each facility. No

\(^3\)This would result in a headway of less than one slot length. However, the duration of this condition would be rather short and hence probably acceptable.
Fig. 42—The analogy between lane-changing and 2-way interchange operation.
Fig. 43--The LINE 1 Corridor Model.
elements were included to model exit storage areas as the simplifying assumption is made that all rates of egress (up to and including 1 vehicle/slot-time) would be possible. (The model is thus simplified as parameters associated with the stochastic exit "service" process are not required.) The end facilities (I and IV) are quite simply modeled by source-sink pairs (SS elements). (Note that the (SG,OZ) pairs, on the approaches to the center interchanges, model the entrance storage areas of the end facilities.)

Vehicle rejection at entrances is modeled as follows:

Vehicles entering the automated system (for example at Facility II) are represented by vehicle entities appearing at the output node of the SS element (B). Subsequently, these entities move to the input node of the SG element, and if the latter is full the entities are simply destroyed — representing rejection.

The required "flow of information" between model elements is as follows. The six OZ elements "feeding" vehicles to each of the two interchanges are grouped into two sets of three OZ elements, and each such set is governed by an interchange scheduler module. Also, there is feedback from each OZ element to the preceding SG element so that the SG gate can be closed when the OZ element reaches its capacity. Maneuvering space reservations for the two-way guideway section, between facilities II and III, are accomplished by information flows between SG elements. When the number of vehicles in SG element 3 is greater than some threshold value the gates for the two upstream SG elements (1 and 2) are closed — hence deferring scheduling of vehicles
at entrances I and II. Thus, as long as maneuvering space is unavailable on the eastbound guideway lane, leading from facility II to III, vehicles entering the corridor at I and II are stored at entrance facilities. Should one of the latter become full, entering vehicles would be rejected. Maneuvering space reservation for the westbound lane from III to II is accomplished by similar information feedback (from SG 4 to SG 5 and SG 6).

Note that the travel time for vehicles going west from a merge of Interchange II to the sink (A), representing egress at Facility I, do so "instantly" as no delay elements are traversed here. An LK element could be inserted here; however, it would not effect the average "extra" delay and flow results obtained from an experimental "run."

2. LINE 2 Model

The simulation model for the LINE 2 Configuration is shown in Fig. 44. Note that two SS elements are used at each of the center facilities (II and III). For example, at Facility II SS-Element B is used to model entering and exiting of westbound vehicles, whereas, SS-Element C accomplishes the same task for eastbound traffic. Thus, the total entering-vehicle flow rate for Facility II is equal to the sum of the source-generation rates for B and C.

The upper and lower portions of the model are completely independent of each other as there are no vehicle entity and/or information flows between same. The SG-threshold feedback for the upper portion is from SG 4 to the gates of 6 and 8 and from 5 to 1 and 3 for the
Fig. 44—The LINE 2 Corridor Model.
lower portion. The OZ elements following SG 2 and SG 4 utilize a com-
mon scheduling module as do those following SG 6 and SG 8. Similarly,
two separate schedulers are used in the eastbound (lower) portion.

3. LINE 3 Model

The LINE 3 Model like the previous one consists of two in-
dependent portions, one for the east- and one for the west-bound
traffic.

Note in Fig. 45 that the models for the swerve areas (along
with the approach lanes to same) are identical to the 2-way interchange
model in Fig. 26. The required SG feedback structure is also the same.
The OZ elements are grouped in six sets (three for each direction of
travel), and thus six independent schedulers are required.

The source and sink common to a given SS element are desig-
nated by the same letter. However, the source-sink pairing is unim-
portant to simulation results as the input and output nodes of an
SS element are completely independent.

Note that there are separate sources for each lane departing
from the end facilities (I and IV). These generate vehicle entities
per the "fixed" routing pattern indicated in Fig. 39. (Note that
unique routes are indicated by the "from" and "to" designations.) For
example, all vehicle entities generated at source A (in Fig. 45) travel
to sink D. This represents the total flow from I to II. In the simple
fixed routing policy considered here, the inside lanes are used as
"through lanes" and the outer ones are used by vehicles that just
entered at the "last facility" or that are to exit at the "next facil-
Fig. 45—The LINE 3 Corridor model.
ity." With this simple procedure all vehicles that pass through a facility do so in the "through lane," thus simplifying the entering process for vehicles coming into the system. In practice, a dynamic routing policy would adjust routing patterns to the demand situation. For example, if there were very few vehicles entering or exiting at the center facilities (II and III), then the traffic between the end facilities (I and IV) would probably be evenly distributed among the available lanes.

C. A Comparative Simulation Study

1. The Model Experiments

The operation of the three corridor configurations can be compared under a variety of both steady-state and time-varying demand conditions. Here, the performance of the three modeled configurations are compared for high-level time-independent demand conditions. The performance measures that were obtained are vehicle flow rates, entrance rejection rates and average vehicle travel delays.  

---

4 This would be accomplished by the central coordinator (mentioned in Chapter III). In the three corridor models presented in this chapter, a central coordinator is not required due to the simple fixed routing policy. Also, the simple entrance rejection policy used does not involve coordination among the various facilities.

5 Vehicle travel delays are calculated by taking actual travel time (source to sink) and subtracting minimum possible travel time. (The latter represents travel time for a vehicle traveling through the system at synchronous speed during its entire trip).
The demand level chosen for testing the LINE 1 and LINE 2 configurations was 1 vehicle/slot-time (at each entrance) — representing a saturation condition;\(^6\) whereas, twice this level was selected for the LINE 3 Configuration (as double-laned guideways were available here). The origin-destination demand matrices were

\[
\begin{array}{cccc}
\text{DESTINATION} & & & \\
I & II & III & IV \\
I & 0 & 1/3 & 1/3 & 1/3 \\
II & 1/3 & 0 & 1/3 & 1/3 \\
III & 1/3 & 1/3 & 0 & 1/3 \\
IV & 1/3 & 1/3 & 1/3 & 0 \\
\end{array}
\]

for LINE 1 and LINE 2 and

\[
\begin{array}{cccc}
\text{DESTINATION} & & & \\
I & II & III & IV \\
I & 0 & 1 & 1/2 & 1/2 \\
II & 1 & 0 & 1/2 & 1/2 \\
III & 1/2 & 1/2 & 0 & 1 \\
IV & 1/2 & 1/2 & 1 & 0 \\
\end{array}
\]

\(^6\)It should be noted here that the exact arrival process is never specified in this dissertation. The assumption is made that vehicles can enter the system only at slot-times with a maximum of 1 vehicle/slot-time for a given entrance storage.
for LINE 3 (both in units of vehicles/slot-time). Note that the demand on the single-laned configurations is balanced, i.e., there is the same rate of origin-destination requests for each O-D pair. The demand chosen for testing the LINE 3 Configuration is balanced except for trips confined to either end (I-II, II-I, III-IV, and IV-III) as these trips are independent of both each other and other possible trips. Thus, rates of 1 vehicle/slot-time could be served, and this demand level was chosen for such trips.

There are several model parameters to be chosen. Although all these parameters are in units of either number of vehicles, slots, or slot-times, it would be useful to assume values for the slot-length and slot-time \( T_s \) so that realistic parameters can be chosen. To this end, a slot-length of 100 ft and slot-time of 1 s, which correspond to \( V_s = 100 \text{ ft/s} \), were selected.\(^7\) Assuming a distance of 1000 ft between the diverge and merge areas of an entrance-exit interchange, an LK element length of 10 slots is appropriate. The LK elements used to model the swerve areas (in LINE 3) were also assumed to be of this length. If the distance between interchanges were 5000 ft, then an OZ length of 50 slots would be appropriate.\(^8\) As OZ elements of the same length are used to schedule entering vehicles, it would take a vehicle a minimum of 50 slot-times (50 s) to enter the system. This choice may

---

\(^7\)These would probably be reasonable values in a future automated highway system [4].

\(^8\)In the LINE 3 Model, the distance between facility II and III is approximately twice that in the other two models as two OZ elements span this distance.
not be realistic; however, it is unimportant as it does not influence the calculated statistics. An OZ capacity of 75 vehicles (1-1/2 times the number of slots) and an SG capacity of 100 were chosen. Thus, on an approach lane to a given interchange, up to 75 vehicles could be in the "scheduled" state and up to 100 vehicles in the deferred state. An SG threshold of 25 vehicles was chosen.9

There is a fixed priority structure for the OZ elements tied to a common scheduler. As previously mentioned, such a structure is used to determine "scheduling service" order when two or more vehicle entities simultaneously appear at the input nodes to their respective OZ elements. As the order-of-arrival priority structure is at a higher priority level, the OZ "service sequence" is rather unimportant and pilot experimentation indicated that results were minimally affected by the sequence for each OZ group. (The priority sequences used are documented in the Appendices.)

Three simulations were conducted — one for each configuration. Each simulation was initiated with an empty network (i.e., no vehicle entities were present in any of the model elements), and was conducted for 10,000 (simulation) slot-times. This corresponded to

9The choice of SG and OZ capacities, and SG threshold are somewhat arbitrary as an "exact" pre-interchange longitudinal maneuvering policy is not presented in this dissertation. (SG, OZ) pairs provide only an approximate model of maneuvering section operation; further refinements are only possible after a maneuvering policy is completely defined.

Results were not obtained for other values of these parameters, as it appeared from pilot experimentation that, at least under the heavy demand conditions considered here, flow rates were only weakly affected by parameter values.
almost 3 hours of real-time operation per the assumption of a 1 s slot-time. The statistical estimates obtained were based on data collected during the whole run length. No replications were considered necessary. The statistical estimators used are described in the DDTS documentation.

2. Corridor Simulation Results

The average flow rates obtained for the three corridor models are shown in Fig. 46. These results are biased slightly to the low side because the simulation models were started in an empty state. For instance, note in the LINE 3 case, that flows which were measured as .99 vehicles/slot-time are known to be 1.00 (per both the demand matrix and the independence of these paths).

The flows obtained here were estimated from vehicle counts at the vehicle entity sinks (inputs to the SS elements). Corresponding origin-destination flow rates are shown in Fig. 47. The average flows on any lane in the network are easily determined as fixed routes are involved, and the origin and destination of each vehicle entity are known.

Vehicle rejection rates are easily estimated from the entering flow rates as the demand at each entrance is known. Entering and exiting flow estimates are shown in Fig. 48. For example, an

---

10 Results obtained for the east- and west-bound halves (of LINE 2 and LINE 3) were almost identical, and as each half is independent of, yet identical to, the other, a single replication was in effect obtained for these configurations.
Fig. 46—Average flow rates in the corridor models.
<table>
<thead>
<tr>
<th>Origin</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td>.20</td>
<td>.19</td>
<td>.20</td>
</tr>
<tr>
<td>II</td>
<td>.19</td>
<td></td>
<td>.21</td>
<td>.19</td>
</tr>
<tr>
<td>III</td>
<td>.21</td>
<td>.20</td>
<td></td>
<td>.21</td>
</tr>
<tr>
<td>IV</td>
<td>.21</td>
<td>.19</td>
<td>.21</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Origin</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td>.20</td>
<td>.20</td>
<td>.20</td>
</tr>
<tr>
<td>II</td>
<td>.33</td>
<td></td>
<td>.29</td>
<td>.30</td>
</tr>
<tr>
<td>III</td>
<td>.29</td>
<td>.30</td>
<td></td>
<td>.33</td>
</tr>
<tr>
<td>IV</td>
<td>.20</td>
<td>.20</td>
<td>.20</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Origin</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td>.99</td>
<td>.37</td>
<td>.36</td>
</tr>
<tr>
<td>II</td>
<td>.99</td>
<td></td>
<td>.37</td>
<td>.36</td>
</tr>
<tr>
<td>III</td>
<td>.37</td>
<td>.37</td>
<td></td>
<td>.99</td>
</tr>
<tr>
<td>IV</td>
<td>.37</td>
<td>.37</td>
<td>.99</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 47—Estimated origin-destination flow rates.](image-url)
entering flow rate of .59 was estimated for Facility I and since the demand level was 1.00, the rejection rate was approximately .41 vehicle/slot-time.

These estimates are also biased slightly due to the (mentioned) flow estimation procedure, and an actual count of rejected vehicles indicated a rate of .38 vehicles/slot-time. In Fig. 49 the entrance rejection rates (here in percents) obtained by entrance counts are shown. The mean rate (averaged over all four entrances) is shown so that the performance of the three configurations may be compared. The reduction from 37.5% to 22.0% with the LINE 2 as compared to the LINE 1 Configuration reflects a considerable improvement in performance.
which must be attributed to the different entrance-exit facilities used. The further reduction in the rejection rate obtained for the LINE 3 Configuration (here a rate of 11.4%) is especially significant as this system was subjected to twice the demand level. The latter improvement is attributed to the use of two lanes in each direction rather than one, and hence a resulting reduction in interactions among maneuvered vehicles.

Average delays (in slot-times) are shown in Fig. 50. Note that a delay of zero slot-times (as for trips from I to II in the LINE 3 Model) indicates that no move-back maneuvers were required, and thus such vehicles traveled at synchronous speed throughout their entire trip. The four O-D pairs (in the LINE 3 Model) where no delay occurred were as expected as there is no interaction here.

Note that most of the delays are quite large. This is a result of the excessive demand levels (as evidenced by the high re-
<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>I</td>
<td>—</td>
<td>139.1</td>
<td>—</td>
<td>133.6</td>
</tr>
<tr>
<td>III</td>
<td>II</td>
<td>180.8</td>
<td>177.5</td>
<td>—</td>
<td>183.2</td>
</tr>
<tr>
<td>IV</td>
<td>III</td>
<td>177.0</td>
<td>173.9</td>
<td>130.3</td>
<td>—</td>
</tr>
</tbody>
</table>

**Line 1**

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>I</td>
<td>—</td>
<td>2.6</td>
<td>—</td>
<td>181.8</td>
</tr>
<tr>
<td>III</td>
<td>II</td>
<td>165.5</td>
<td>166.6</td>
<td>—</td>
<td>203.2</td>
</tr>
<tr>
<td>IV</td>
<td>III</td>
<td>195.4</td>
<td>192.8</td>
<td>186.1</td>
<td>—</td>
</tr>
</tbody>
</table>

**Line 2**

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>I</td>
<td>—</td>
<td>0</td>
<td>166.8</td>
<td>166.6</td>
</tr>
<tr>
<td>III</td>
<td>II</td>
<td>0</td>
<td>—</td>
<td>161.4</td>
<td>165.5</td>
</tr>
<tr>
<td>IV</td>
<td>III</td>
<td>169.8</td>
<td>169.8</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>162.9</td>
<td>168.4</td>
<td>0</td>
<td>—</td>
</tr>
</tbody>
</table>

**Line 3**

Fig. 50—Average delays (in slot-times) for the Corridor Models.
jection rates) used in this comparative study. The delays would have been even larger if larger capacity OZ and SG elements had been used. Delay statistics are studied in more detail in the next chapter where loop networks are examined under moderate demand levels.

D. Model Utility

The corridor models are useful as both conceptual tools and for simulation (experimentation) studies. If the models adequately represent the operation of the conceived corridor configurations, then they should be useful in predicting the performance of same.

It has probably been noted that these models do not "depict" the fine details of system operation. For example, the detailed movement of vehicles through a longitudinal maneuvering section is not explicitly defined in the models, but rather a "black box" approach to maneuvering section modeling is used. (Vehicle entities entering and departing an (SG,OZ) pair can be observed, but little can be ascertained about their trajectories within the maneuvering section represented.)

In this context, it may be helpful to indicate several reasons for using simplified (abstract) models — among these are:

(1) Many details associated with microscopic operation are yet unknown, that is, the systems modeled are only partially specified;
(2) If "too much" detail were included, then a model would tend to lose its utility as a conceptual or communication tool;

(3) The models are probably adequate for estimating the performance measures of interest here, and more detailed models may not be cost-effective.
CHAPTER VI
LOOP NETWORKS

A loop network is essentially a corridor network constructed along a circular right-of-way and thus the operation of the two should be similar. Further, in both cases the only interchanges required are those at entrance-exit facilities.

In this chapter, two different loop configurations are examined, and for one of these, delay statistics are obtained for various demand levels, loop sizes (number of facilities on the loop), and trip lengths.

A. Alternative Configurations

The two representative loop networks -- LOOP 1 and LOOP 2 -- shown in Fig. 51, utilize entrance-exit facilities corresponding to those in LINE 1 and LINE 2 (the corridor networks), respectively. In each configuration there is a clockwise (c.w.) and a counter-clockwise (c.c.w.) guideway loop. Thus there are two routes between each facility, e.g., a vehicle traveling from I to III may do so either on the c.w.-loop via Facility II or on the c.c.w.-loop via Facility IV. Note that in the LOOP 1 Configuration, the c.w. and the c.c.w. loops are interconnected by 3-way interchanges; whereas, in the LOOP 2 Network
Fig. 51—Representative loop networks.
these guideway loops are not connected. Per the discussion and results for the corresponding corridor configurations (LINE 1 and LINE 2), superior performance would be expected for the LOOP 2 over the LOOP 1 Configuration.¹

B. Loop Models

1. LOOP 1 Model

The 4-facility LOOP 1 Configuration would be modeled by connecting four of the "E-E Models," shown in Fig. 29, to form a closed loop. For experimentation, however, a model for a 5-facility configuration was constructed as such would provide a unique "shortest route" between each OD pair. (All inter-facility distances are assumed to be equal.) The routes for vehicles originating at Facility I are depicted in Fig. 52.

The following parameter values were selected for the 5-facility LOOP 1 model:

- **LK Length** = 1 slot;
- **OZ Length** = 5 slots;
- **OZ Capacity** = 8 vehicles;
- **SG Threshold** = 3 vehicles; and
- **SG Capacity** = 24 vehicles.

These parameter values, although not realistic, were chosen as such a "scaled down" network clearly illustrates a phenomenon that shall be

¹A multiple-laned loop network corresponding to the LINE 3 Network is not considered here.
Fig. 52—Routing patterns for a 5-Facility LOOP 1 Model.

termed "shutdown." The latter is defined as an abnormal cessation of flows on the network — that results in vehicles being trapped within same. A shutdown event is expected to result when each interchange controller cannot obtain maneuvering space (for vehicles it serves) on downstream approaches to following interchanges, and hence each interchange controller must bring all vehicles on its approach lanes to a stop. Vehicles cannot be permitted to continue as this would "violate" maneuvering space limitations on the approach to the next interchange.

In the LOOP 1 model a shutdown is represented by the concurrent closure of all SG-element gates as a result of SG-to-SG (threshold) feedback. Each SG element in the closed loop is upstream to itself, and once all SG elements in a loop are closed, they will be
permanently locked in that state. The entities contained within the SG elements on the loop represent vehicles which would be "trapped" in the deferred state.

A single simulation run was conducted for 10,000 slot-times starting with an empty network. A balanced time-varying demand matrix was utilized, wherein, the demand rate was increased linearly from 0.0 to 1.0 vehicles/slot-time over the first half of the simulation run and then held constant for the remaining portion. (Such a demand could represent the onset of a "rush hour.")

Both the entering and the (c.w. and c.c.w.) on-loop flow densities (vehicles/slot),\(^2\) as measured at one of the five facility interchanges, are shown in Fig. 53. These flows are determined from slot-occupancy at the output nodes of the OZ elements, and are plotted as a function of time. Note that the entering flow increased approximately with the time-varying demand level until \(t = 5000\) at which time\(^3\) flow began to decrease, and at \(t = 4700\) shutdown had clearly occurred — as all flows were zero from this time to the end of the run. Due to both network and demand symmetry, it is not surprising that flows at the other four facilities were found to be quite similar.

One important result verified by the study of the LOOP 1 model is that shutdown situations can occur, and hence must be avoided. A suggested procedure to accomplish same would be to meter entrance

\(^2\)Flow densities (vehicles/slot) and flow rates (vehicles/slot-time) are identical measures.

\(^3\)Time is measured in slot-times.
Fig. 53—Flow densities for the 5-facility LOOP 1 Model.
flows so that the probability of a shutdown is small. Should shutdown occur the "deadlock" could be resolved by requiring a limited number of vehicles to leave the system. (They could attempt reentry if they had not reached their destination, and could possibly be given priority over other entering vehicles.)

2. LOOP 2 Model

In the LOOP 2 Configuration, the operation of the c.w. and c.c.w. loops are both independent and identical, and hence only a single direction need be modeled such as the 4-facility c.w. one shown in Fig. 54. Operation is similar to that of the LINE 2 Configuration. Four independent schedulers are required — one being associated with the pair of OZ elements on the approach to each facility (e.g., the OZ elements following SG 2 and SG 5). The information feedback paths associated with SG 6 are shown, and note that this involves SG threshold feedback to the two upstream elements (SG 2 and SG 5) and also the OZ-to-SG feedback. As feedback paths from SG 6 to 5 to 8 to 7 and back to 6 would form a closed loop, shutdown would also be possible with this configuration.

Both 4-facility and 8-facility single-loop models were evaluated under a variety of demand conditions. In every case a

4 However, the demands on these two portions of the configuration are generally related.

5 The following parameter values were selected:
   LK Length = 1 slot;
   OZ Length = 25 slots;
   OZ Capacity = 40 vehicles;
   SG Threshold = 20 vehicles; and
   SG Capacity = 60 vehicles.
Fig. 54—The clockwise-loop (model) of the 4-facility LOOP 2 Configuration.
balanced demand matrix was used,\(^6\) and the level of demand chosen to result in a pre-specified on-loop flow density. A constant demand level was used over each 10,000 slot-time run and in no case did a shutdown occur. (Other demand matrices or longer runs would have possibly resulted in shutdowns.)

Both average and standard-deviation delay statistics are plotted as a function of the on-loop flow density in Figs. 55 and 56, respectively. Note that both statistics increase with flow density (at possibly an exponential rate for flows less than .80), and also that the 8-facility loop has both higher average delay and higher standard deviation of delay than the 4-facility loop (for a given flow density). Thus, both the expected travel time and its dispersion increase with the demand level and with the number of facilities on the loop.

It is interesting to note that in all cases both average and standard deviation of delay increase with trip length as indicated by results for the 4-facility (Figs. 57 and 58) and 8-facility (Figs. 59 and 60) models. Here delay statistics are plotted as a function of on-loop flow density with trip length (designated by the number of inter-facility link distances traveled) as a parameter. These results are as expected as a vehicle travels through more merges (and hence has more maneuvering delays and dispersion in same) for longer trips. It is interesting to note that if no maneuvering were allowed on the guideway

\(^6\)Note that this does not correspond to a "shortest route" policy for the complete network (c.w. and c.c.w. loops combined).
Fig. 55—Average delay versus on-loop flow density for the LOOP 2 Models.
Fig. 56—Standard-deviation of delay versus on-loop flow density for the LOOP 2 Models.
Fig. 57—Average delays for various trip lengths in the 4-facility LOOP 2 Model.
Fig. 58--Standard deviation of delay for various trip lengths in the 4-facility LOOP 2 Model.
Fig. 59—Average delays for various trip lengths in the 8-facility LOOP 2 Model.
Fig. 60—Standard deviation of delay for various trip lengths in the 8-facility LOOP 2 Model.
loop (and thus all delays would occur to vehicles in the entrance storages), the delay statistics would be almost insensitive to trip length (unless separate entrance queues were provided for various destinations). However, in the strategy used here travelers taking long trips would incur more maneuvering delay than those on short trips.

To demonstrate the effect of an egress limitation, one facility (I) in the LOOP 2 model was modified per Fig. 61. Here, two consecutive merges were used to represent an egress flow restriction.

![Fig. 61—Facility I with appropriate additions for modeling egress limitations.](image-url)
Exiting vehicle entities were required to schedule through these merges, and the rate of egress was controlled via the rate of entity generation at E and F. The OZ elements a, b, and c were tied to a common scheduler which was operated per an order-of-arrival procedure.

Two cases were simulated and flow densities were measured at a representative facility (II). In the first case no egress flow limitation was imposed; whereas, in the second one an anomalous egress restriction was simulated by a step increase in the entity generation rates at E and F. (These were both increased from 0 to .45 vehicles/slot-time at $t = 3000$.) Results for the first case are shown in Fig. 62 and those for the second in Fig. 63. Note that results were identical up to $t = 3000$ after which shutdown occurred in the second case.

After $t = 3000$ the average "exit-service" rate decreased from 1 vehicle/slot-time to only 1/3 vehicle/slot-time; whereas, the demand pattern indicated that a .45 vehicle/slot-time exit rate was required to meet the demand situation. In the simulation, the immediate result of the egress restriction was a rapid increase in the number of vehicles in the SG element which represents the exit storage area. However, as the exit could not accommodate the egress flow the SG-to-SG (threshold) feedback shown in Fig. 61 resulted in a restriction in mainline flow. Subsequently, shutdown resulted via information

---

7In both cases a balanced demand was used and the demand level was chosen to result in an on-loop flow of .90 vehicles/slot.
Fig. 62—Flows at Facility II for case with no egress restrictions.
Fig. 63—Flows at Facility II for case with an egress restriction at Facility I at $t = 3000$. 

FLOW DENSITY (veh/Slot) vs simulation TIME (slot times)
feedback around the loop.

In practice, a shutdown could be avoided by limiting entrance flows and/or rerouting vehicles to the next facility (following the one impeded by an anomalous restriction).
A. Introduction

The urban part of an intercity, dual-mode, automated transportation system would serve both as an interface with conventional city freeways and streets, and as an inter-city transportation mode. Such an urban network would probably include interchanges at other than entrance-exit facilities, per the representative network shown in Fig. 64. Here there is an inner and outer loop (ring) and four radial corridors. The latter are connected to the inner loop via 3-way interchanges and to the outer loop via 4-way interchanges. Note that this network is depicted as surrounding the central business district -- which would possibly be served by an automated captive-vehicle system. This network includes 40 interchanges with 32 of these at entrance-exit facilities.

It seemed more effective to consider the simplified network shown in Fig. 65 as it is both considerably simpler and yet contains the essential attributes of the network in Fig. 64. This network is comprised of an inner ring, an outer ring, and four radials and thus several viable routes are available between each O-D pair. Note that its ten entrance-exit facilities are designated by the letters A through J.
Fig. 64—A representative urban network.
Fig. 65—The urban network configuration studied.
B. Routing Policies

In the previously considered networks, simple routing policies were possible due to the limited choice of paths between each 0-D pair; however, here the choice of several suitable paths between each 0-D pair, complicates routing decisions. To illustrate the possible paths, a simplified representation of the urban network is shown in Fig. 66. Here, the lines represent bidirectional guide—

Fig. 66—A simplified representation of the urban network.
Fig. 67—Some possible paths from A to J, and from G to C.
way links, the intersections of same represent either 3-way or 4-way interchanges, and the letters represent entrance-exit points. Clearly, a single "best" route exists for many of these O-D pairs as other routes would result in considerably longer travel times. For example, a single "direct" route is available between adjacent entrance-exit points on the outer ring (such as G and H or H and I), and between radial pairs such as G and A or B and C. For other pairs such as A and B, two "direct" routes are available with the choice between these depending on the inner-ring traffic conditions. For more distantly separated pairs such as A and J or G and C, many viable routes exist as illustrated in Fig. 67.

Routing policies can be classified as either demand responsive or demand independent. In the latter, the route chosen for a given vehicle is dependent only on the origin and destination of the vehicle, i.e., a "fixed" route exists for each O-D pair; whereas in the former, route choice is dependent on the traffic demand pattern. Routing policies can also be classified according to whether the route for a vehicle is determined prior or post to its dispatch from an entrance area. With a pre-dispatch guideway allocation scheduling procedure (see Chapter II) only pre-dispatch routing policies are generally feasible; whereas, with a post-dispatch guideway allocation

\[^{1}\text{However, in a system employing pre-dispatch procedures for scheduling and routing, post-dispatch modes (procedures) are clearly necessary for handling many types of anomalous events (e.g., those resulting in a lane blockage).}\]
scheme (e.g., the dynamic scheduling policy developed in this dissertation), post-dispatch routing decisions can easily and effectively be made and implemented.

A post-dispatch responsive routing policy will be termed a dynamic routing policy. Such schemes would have the following advantages over pre-dispatch routing policies:

1. Vehicles can be easily rerouted around anomalous blockages; and
2. Unexpected fluctuations in short trip demands can be generally accommodated by selective routing of vehicles on long trips.\(^2\)

In the controller architecture presented in Chapter III, the routing function was assumed to be part of the central coordinating task. The decision as to the direction to be followed by each vehicle at an interchange diverge would be made by a local controller on the basis of control information from the central coordinator. The central coordinator would specify the percentage of vehicles which would follow various routes and thus two vehicles traveling to the same destination and entering an interchange on the same approach lane could depart from the interchange in different directions.

\(^2\)For example, should there be an unexpectedly high demand from A to B on the inner loop (see Fig. 66), vehicles on long intercity trips (and traveling through this urban area) would probably be routed on the outer loop rather than via the radials and inner loop.
For example, at a given time the central coordinator might specify that 40% of the C-bound vehicles entering the 4-way interchange on the outer loop at G, travel clockwise on the outer loop, and the remaining 60% travel on the c.c.w. outer-loop path (with no vehicles traveling via the radial path). However, when each C-bound vehicle reaches the 4-way interchange (at either I or E) another routing decision could be made to determine whether it continued on the outer loop or traveled via the radials and inner loop. The fraction of vehicles using each path could be determined by the anticipated traffic on each.

C. The Urban Network Model

A simulation model of the urban network under consideration is presented in Fig. 68. A simple demand-independent (fixed) routing procedure was used in this model. The direction in which a vehicle entity leaves an interchange is determined by its destination per the designations shown at each interchange submodel in Fig. 68. For example, a vehicle entity generated at G follows the c.w. path to destination H, I, or J, the c.c.w. path for D, E, and F, and the

---

3As the focus of this dissertation is the first investigation of the dynamic scheduling technique, it was considered beyond the desired scope to evaluate dynamic routing policies.
Fig. 68—An urban network model.
radial path for A, B, and C. The assigned routes for vehicles originating at an inner-ring facility (A) (and going to one of the nine other facilities) are shown in Fig. 69; whereas routes originating at outer-ring O-D points (here, represented by C and D) are shown in Fig. 70.

Fig. 69—Paths originating at one of the inner loop facilities (A).
Fig. 70—Paths originating at C and D.
The network model, shown in Fig. 68, contains six 3-way interchange submodels and six 4-way ones as detailed in Figs. 27 and 28, respectively. Since the approach lanes to each interchange are included in the interchange models, no other simulation elements are needed to model the inter-interchange guideway segments. The connections of the submodels are indicated by the numbered lines, and each source-sink pair (SS element) is denoted by a circled letter (e.g., the SS-element represented by H along with the 3-way interchange submodel to which it is connected are modeled as shown in Fig. 29.)

The complete model consists of a total of 334 basic model elements as follows:

- 10 SS elements;
- 66 DG elements;
- 66 HG elements;
- 42 SG elements;
- 42 OZ elements; and
- 108 LK elements.

Both OZ-to-SG feedback and SG-to-SG (threshold) feedback are used per the general discussion in Chapter III. A total of 42 OZ-to-SG feedback paths are required as there are 42 (SG,OZ) pairs and 82 SG-to-SG information paths are required.

If the model were subjected to a balanced demand with an
average rate of $\lambda$ vehicle/slot-time between each O-D pair, both the total entrance flow and total exit flow at a given facility would be $9 \lambda$. With the particular routing policy chosen, the average c.w. flow rate at any point on the outer loop would be $7 \lambda$, while the expected c.c.w. rate would be $9 \lambda$. The flow rate on the radials between the inner and outer loops would be $5 \lambda$ in each direction (in each of the four radials), and inner loop flow rates would be $5 \lambda$ and $4 \lambda$ in the c.w. and c.c.w. directions, respectively. The total average rate of vehicle flow into (or out of) each interchange is as shown in Fig. 71. Note that the interchanges on the inner loop are subjected to lighter demand than those on the outer loop.

D. Model Experimentation

The following set of parameters were chosen for model experimentation:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LK Length</td>
<td>7 slots</td>
</tr>
<tr>
<td>OZ Length</td>
<td>100 slots</td>
</tr>
<tr>
<td>OZ Capacity</td>
<td>150 vehicles; and</td>
</tr>
<tr>
<td>SG Threshold</td>
<td>50 vehicles; and</td>
</tr>
<tr>
<td>SG Capacity</td>
<td>200 vehicles.</td>
</tr>
</tbody>
</table>

4No round trips (e.g., A to A) are included in this demand, and thus the demand matrix has zeroes on the diagonal.

5In the 4-way interchanges some links have lengths of only 5 slots to equalize "through-interchange" travel times (see Chapter IV).
Various model parameters could be utilized for each interchange submodel; however, the use of this "universal" set simplified the task of model construction. Also, for moderate levels of demand
(\(\lambda \leq 0.09\)), delay and flow statistics are expected to be insensitive
to the set of parameters chosen (as required maneuvering delays are
expected to be small).

Six simulations were conducted for 10,000 slot-times each
(starting with an empty network in all cases). Balanced time-indepen­
dent demands with \(\lambda = 0.050, 0.065, 0.080, 0.090, 0.091, \) and \(0.093\)
vehicles/slot-time were used corresponding to entering flow rates of
0.450, 0.585, 0.720, 0.810, 0.819, and 0.837 vehicles/slot-time, respectively
(assuming no rejections occur).

Similar flow rate and delay data were collected for each
demand level and the majority of the results to be presented here
are for one case, \(\lambda = 0.065\) — as this case is representative of the
lower-demand ones (\(\lambda \leq 0.09\)). The number of completed trips between
each 0-D pair are tabulated in Fig. 72 for the \(\lambda = 0.065\) case — for
which approximately 650 trips are expected between each 0-D pair.
Most of the matrix entries are slightly smaller than this — probably
a result of starting the network in an empty state. No rejections occurred
for this case as can also be noted from Fig. 72.

Flow rates were measured at each of the twelve interchanges,
and measurements\(^6\) for the representative interchanges \(\alpha, \beta, \gamma, \delta, \) and \(\epsilon\)
(designated in Fig. 73) and the case \(\lambda = 0.065\) are plotted in Figs. 74-78.

\(^6\)The measurement points are represented by numerals in Fig. 73.
In the simulation model, flows were obtained by time averages (taken
over 300 slot-times) of slot occupancy at the output nodes of the OZ ele­
ments. Estimates are shown at 100 slot-time intervals and hence there is
some overlap. Flows are in units of percent-of-slot-occupancy.
### NO. OF TRIPS MATRIX

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>644</td>
<td>640</td>
<td>564</td>
<td>617</td>
<td>610</td>
<td>652</td>
<td>583</td>
<td>629</td>
<td>654</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>600</td>
<td>0</td>
<td>594</td>
<td>600</td>
<td>644</td>
<td>653</td>
<td>613</td>
<td>667</td>
<td>628</td>
<td>673</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>591</td>
<td>630</td>
<td>0</td>
<td>650</td>
<td>639</td>
<td>621</td>
<td>648</td>
<td>591</td>
<td>652</td>
<td>649</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>603</td>
<td>647</td>
<td>614</td>
<td>0</td>
<td>664</td>
<td>629</td>
<td>618</td>
<td>646</td>
<td>594</td>
<td>630</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>648</td>
<td>594</td>
<td>680</td>
<td>638</td>
<td>0</td>
<td>590</td>
<td>618</td>
<td>609</td>
<td>566</td>
<td>642</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>406</td>
<td>670</td>
<td>608</td>
<td>651</td>
<td>700</td>
<td>0</td>
<td>641</td>
<td>654</td>
<td>654</td>
<td>592</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>667</td>
<td>607</td>
<td>591</td>
<td>561</td>
<td>644</td>
<td>678</td>
<td>0</td>
<td>673</td>
<td>677</td>
<td>679</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>541</td>
<td>550</td>
<td>610</td>
<td>622</td>
<td>588</td>
<td>648</td>
<td>643</td>
<td>0</td>
<td>675</td>
<td>672</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>621</td>
<td>600</td>
<td>648</td>
<td>661</td>
<td>610</td>
<td>673</td>
<td>596</td>
<td>641</td>
<td>0</td>
<td>604</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>628</td>
<td>642</td>
<td>630</td>
<td>660</td>
<td>631</td>
<td>572</td>
<td>678</td>
<td>673</td>
<td>636</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### NO. OF REJECTS

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 72—The number-of-completed-trips matrix (case $\lambda = .065$).
Fig. 73—Flow measurement points in one quadrant of the urban model.

In all cases the average flows are consistent with expected results, and the interchanges all appear to reach steady-state flow levels after only a short initialization time (1000 slot-times). (However, it is thought that delay statistics might take considerably longer to reach steady-state values [36]).
Fig. 74—Flow rates at Interchange a (case $\lambda = .065$).
Fig. 75—Flow rates at Interchange $\beta$ (case $\lambda = 0.065$).
Fig. 76—Flow rates at Interchange $\gamma$ (case $\lambda = 0.065$).
Fig. 77—Flow rates at Interchange $\delta$ (case $\lambda = .065$).
Fig. 78--Flow rates at Interchange $\epsilon$ (case $\lambda = 0.05$).
The minimum measured travel time for each O-D pair (for \( \lambda = .065 \)) are shown in Fig. 79. These results indicate that at least one vehicle traveled between each O-D pair with no delay maneuvers (i.e., at synchronous speed throughout its entire trip).* The corresponding mean, max and standard deviation of travel time matrices are shown in Figs. 80-82. (All results are in slot-times.)

Mean (or average) delay statistics are calculated from the mean travel times by subtracting synchronous travel time. The average and standard deviation of delay obtained for each demand case are plotted in Fig. 83. These statistics (obtained from delay data collected for all completed trips) indicate that the urban network "saturates" at a \( \lambda \) of approximately .090 vehicles/slot-time as delays increase markedly for demands greater than this.

In Figs. (84) and (85) the delay statistics (mean and standard-deviation) are plotted with trip length as a parameter. (The latter is specified by the number of interchanges a vehicle traverses on its trip.) Note that each of these statistics increase with both the demand level (\( \lambda \)) and with the trip length -- as might be expected. For \( \lambda < .080 \) both statistics appear to increase exponentially with each parameter; whereas, for \( \lambda > .080 \) a super-exponential increase is apparent.

According to flow measurement data obtained at each interchange, a shutdown occurred only in the two higher-demand cases, \( \lambda = .091 \) and \( \lambda = .093 \) -- with approximate shutdown times (defined here as the time at which all vehicle entity flows cease) of \( t = 6000 \)

---

*This result was expected since the network was initialized in an empty state.
### U/O MIN-TRAVEL-TIME MATRIX

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0</td>
<td>330.0</td>
<td>440.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>330.0</td>
<td>330.0</td>
<td>440.0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>330.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>330.0</td>
<td>440.0</td>
<td>440.0</td>
<td>440.0</td>
<td>330.0</td>
<td>330.0</td>
</tr>
<tr>
<td>C</td>
<td>440.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
</tr>
<tr>
<td>D</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
</tr>
<tr>
<td>E</td>
<td>330.0</td>
<td>330.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
</tr>
<tr>
<td>F</td>
<td>330.0</td>
<td>440.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
</tr>
<tr>
<td>G</td>
<td>220.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
</tr>
<tr>
<td>H</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
<td>330.0</td>
<td>220.0</td>
</tr>
<tr>
<td>I</td>
<td>330.0</td>
<td>330.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
<td>220.0</td>
</tr>
<tr>
<td>J</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>330.0</td>
<td>440.0</td>
<td>550.0</td>
<td>440.0</td>
<td>330.0</td>
<td>220.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Fig. 79—The minimum-travel-time matrix (case \( \lambda = 0.065 \)).
### G/U Mean-travel-time matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0</td>
<td>330.7</td>
<td>441.5</td>
<td>442.7</td>
<td>331.1</td>
<td>331.0</td>
<td>271.0</td>
<td>333.5</td>
<td>330.9</td>
<td>441.0</td>
</tr>
<tr>
<td>B</td>
<td>330.0</td>
<td>0.0</td>
<td>220.9</td>
<td>331.5</td>
<td>330.9</td>
<td>441.4</td>
<td>441.5</td>
<td>442.3</td>
<td>331.2</td>
<td>332.1</td>
</tr>
<tr>
<td>C</td>
<td>441.6</td>
<td>331.2</td>
<td>0.0</td>
<td>270.9</td>
<td>331.5</td>
<td>441.9</td>
<td>552.3</td>
<td>443.9</td>
<td>333.0</td>
<td>221.7</td>
</tr>
<tr>
<td>D</td>
<td>442.1</td>
<td>332.4</td>
<td>221.8</td>
<td>0.0</td>
<td>221.2</td>
<td>331.7</td>
<td>442.3</td>
<td>555.0</td>
<td>444.0</td>
<td>332.5</td>
</tr>
<tr>
<td>E</td>
<td>331.7</td>
<td>331.4</td>
<td>332.9</td>
<td>221.6</td>
<td>0.0</td>
<td>220.9</td>
<td>331.5</td>
<td>442.1</td>
<td>557.0</td>
<td>444.1</td>
</tr>
<tr>
<td>F</td>
<td>442.6</td>
<td>444.1</td>
<td>332.7</td>
<td>221.6</td>
<td>0.0</td>
<td>221.2</td>
<td>331.7</td>
<td>442.4</td>
<td>554.8</td>
<td>554.8</td>
</tr>
<tr>
<td>G</td>
<td>220.9</td>
<td>441.2</td>
<td>557.3</td>
<td>443.8</td>
<td>332.8</td>
<td>221.6</td>
<td>0.0</td>
<td>221.0</td>
<td>331.5</td>
<td>442.1</td>
</tr>
<tr>
<td>H</td>
<td>332.0</td>
<td>441.8</td>
<td>442.3</td>
<td>554.9</td>
<td>443.9</td>
<td>332.4</td>
<td>221.7</td>
<td>0.0</td>
<td>221.3</td>
<td>331.0</td>
</tr>
<tr>
<td>I</td>
<td>330.8</td>
<td>331.0</td>
<td>331.5</td>
<td>442.1</td>
<td>551.9</td>
<td>443.7</td>
<td>332.9</td>
<td>221.6</td>
<td>0.0</td>
<td>221.0</td>
</tr>
<tr>
<td>J</td>
<td>442.0</td>
<td>331.9</td>
<td>221.7</td>
<td>331.8</td>
<td>442.3</td>
<td>554.7</td>
<td>444.0</td>
<td>332.6</td>
<td>221.6</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

**Fig. 80**—The mean-travel-time matrix (case \[ \lambda = .065 \]).
### 0/0 MAX-TRAVEL-TIME MATRIX

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0</td>
<td>334.0</td>
<td>448.0</td>
<td>449.0</td>
<td>340.0</td>
<td>341.0</td>
<td>278.0</td>
<td>340.0</td>
<td>330.0</td>
<td>448.0</td>
</tr>
<tr>
<td>B</td>
<td>335.0</td>
<td>0.0</td>
<td>226.0</td>
<td>338.0</td>
<td>358.0</td>
<td>449.0</td>
<td>448.0</td>
<td>449.0</td>
<td>337.0</td>
<td>339.0</td>
</tr>
<tr>
<td>C</td>
<td>452.0</td>
<td>230.0</td>
<td>0.0</td>
<td>227.0</td>
<td>341.0</td>
<td>450.0</td>
<td>560.0</td>
<td>453.0</td>
<td>341.0</td>
<td>229.0</td>
</tr>
<tr>
<td>D</td>
<td>448.0</td>
<td>342.0</td>
<td>231.0</td>
<td>0.0</td>
<td>728.0</td>
<td>338.0</td>
<td>450.0</td>
<td>566.0</td>
<td>455.0</td>
<td>342.0</td>
</tr>
<tr>
<td>E</td>
<td>327.0</td>
<td>339.0</td>
<td>343.0</td>
<td>228.0</td>
<td>0.0</td>
<td>226.0</td>
<td>338.0</td>
<td>451.0</td>
<td>558.0</td>
<td>457.0</td>
</tr>
<tr>
<td>F</td>
<td>337.0</td>
<td>452.0</td>
<td>454.0</td>
<td>342.0</td>
<td>230.0</td>
<td>0.0</td>
<td>229.0</td>
<td>340.0</td>
<td>450.0</td>
<td>567.0</td>
</tr>
<tr>
<td>G</td>
<td>226.0</td>
<td>449.0</td>
<td>338.0</td>
<td>342.0</td>
<td>228.0</td>
<td>0.0</td>
<td>226.0</td>
<td>341.0</td>
<td>450.0</td>
<td>567.0</td>
</tr>
<tr>
<td>H</td>
<td>334.0</td>
<td>450.0</td>
<td>451.0</td>
<td>338.0</td>
<td>454.0</td>
<td>341.0</td>
<td>278.0</td>
<td>0.0</td>
<td>227.0</td>
<td>338.0</td>
</tr>
<tr>
<td>I</td>
<td>338.0</td>
<td>339.0</td>
<td>339.0</td>
<td>450.0</td>
<td>338.0</td>
<td>453.0</td>
<td>343.0</td>
<td>329.0</td>
<td>0.0</td>
<td>228.0</td>
</tr>
<tr>
<td>J</td>
<td>448.0</td>
<td>342.0</td>
<td>227.0</td>
<td>338.0</td>
<td>449.0</td>
<td>567.0</td>
<td>453.0</td>
<td>343.0</td>
<td>228.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Fig. 81**—The max-travel-time matrix (case $\lambda = .065$).
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.00</td>
<td>1.13</td>
<td>1.42</td>
<td>1.80</td>
<td>1.76</td>
<td>1.65</td>
<td>1.76</td>
<td>1.57</td>
<td>1.06</td>
<td>1.50</td>
</tr>
<tr>
<td>B</td>
<td>0.99</td>
<td>0.00</td>
<td>1.11</td>
<td>1.49</td>
<td>1.12</td>
<td>1.35</td>
<td>1.50</td>
<td>1.78</td>
<td>1.74</td>
<td>1.78</td>
</tr>
<tr>
<td>C</td>
<td>1.56</td>
<td>1.53</td>
<td>0.00</td>
<td>1.20</td>
<td>1.66</td>
<td>1.71</td>
<td>1.92</td>
<td>2.53</td>
<td>2.02</td>
<td>1.68</td>
</tr>
<tr>
<td>D</td>
<td>1.70</td>
<td>2.20</td>
<td>1.80</td>
<td>0.00</td>
<td>1.79</td>
<td>1.53</td>
<td>1.97</td>
<td>2.94</td>
<td>2.61</td>
<td>2.14</td>
</tr>
<tr>
<td>E</td>
<td>1.55</td>
<td>1.40</td>
<td>2.18</td>
<td>1.55</td>
<td>0.00</td>
<td>1.15</td>
<td>1.52</td>
<td>1.97</td>
<td>1.64</td>
<td>2.86</td>
</tr>
<tr>
<td>F</td>
<td>1.50</td>
<td>1.90</td>
<td>2.77</td>
<td>2.16</td>
<td>1.59</td>
<td>0.00</td>
<td>1.40</td>
<td>1.71</td>
<td>2.00</td>
<td>2.68</td>
</tr>
<tr>
<td>G</td>
<td>1.03</td>
<td>1.38</td>
<td>1.80</td>
<td>2.41</td>
<td>2.04</td>
<td>1.41</td>
<td>0.00</td>
<td>1.58</td>
<td>1.63</td>
<td>1.91</td>
</tr>
<tr>
<td>H</td>
<td>1.72</td>
<td>1.67</td>
<td>1.97</td>
<td>2.87</td>
<td>2.63</td>
<td>1.90</td>
<td>1.67</td>
<td>0.00</td>
<td>1.39</td>
<td>1.64</td>
</tr>
<tr>
<td>I</td>
<td>1.15</td>
<td>1.35</td>
<td>1.54</td>
<td>1.64</td>
<td>1.71</td>
<td>2.49</td>
<td>2.12</td>
<td>1.63</td>
<td>0.00</td>
<td>1.20</td>
</tr>
<tr>
<td>J</td>
<td>1.63</td>
<td>1.79</td>
<td>1.31</td>
<td>1.67</td>
<td>1.86</td>
<td>2.84</td>
<td>2.61</td>
<td>2.15</td>
<td>1.52</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Fig. 82—The standard-deviation-of-travel-time matrix (case $\lambda = .065$).
Fig. 83—Delay statistics for the urban model subject to a balanced demand.
Fig. 84—Average delay as a function of the demand level ($\lambda$) with trip length as a parameter.
Fig. 85—Standard-deviation of delay as a function of the demand level ($\lambda$) with trip length as a parameter.
and $t = 8500$, respectively. Also, entrance rejection only occurred for the two higher-demand cases, wherein all entering vehicles were rejected after shutdown. If the model parameters had been chosen such that maneuvering space was quite limited, it is expected that some rejections would have also occurred for the lower-demand cases ($\lambda < .090$), and further, shutdown would probably be more likely to occur. Also, it is expected that even with the model parameters chosen, shutdown would occur with some of the lower-demand cases if sufficiently long runs were conducted.

It appears that in practice, shutdowns could be prevented (with a high probability) by the inclusion of two additional operations: entrance-flow metering, and forced exiting. The latter would only be necessary should a shutdown be imminent or have occurred.

It is worth comparing the per lane flow rates measured here with those typically obtained in conventional networks. In the latter, the flows are below some 2000 vehicles/lane/hour (with this value only being achieved on limited-access highways under nearly ideal flow conditions). Here, if it were assumed that the maximum usable (balanced) demand level were $\lambda = 0.09$ veh/slot time, then for a slot time of 1 sec., the per lane flow would be as great as 2916 veh/lane/hr. If it were technologically feasible to choose $T = 0.5$ sec., then the flow would be as high as 5832 veh/lane/hr. Since it is quite possible that greater flows could be achieved via other routing policies, automated networks, such as the one considered here, clearly have considerable potential for improving the vehicle-throughput efficiency of roadway networks.
This chapter contains a summary of this research, a set of final conclusions, and ideas for further research.

A. Summary

One proposed class of partial solutions to today's ground transportation problems is the development of automated vehicle systems for both urban and inter-city use, and three types of such systems have been proposed — pallet-vehicle, captive-vehicle, and dual-mode-vehicle systems. Many general feasibility studies have been undertaken, while other studies have dealt specifically with various microscopic and macroscopic control problems associated with automated vehicle systems. The former include problems associated with individual vehicle position-regulation and -maneuvering while the latter includes problems of system supervision, coordination and control. The chief control task would be the coordination of vehicle movement such that each would be routed from its origin to its destination in an efficient, safe, and conflict-free fashion.

The objective of the research reported here was the development of a macroscopic control approach suited to the task of
scheduling and routing dual-mode or automated highway vehicles
(although most results were expected to apply directly to other
types of automated vehicle systems).

First, a taxonomy tree of macroscopic control approaches
was constructed and used to classify control algorithms used in
prior studies, and three approaches were illustrated using a simple
network. The approaches considered incorporated a roadway (guideway)
based, moving-position reference (these are commonly termed "syn­
chronous", "slot", or "moving-cell" approaches), and here, such
were subclassified according to whether guideway space would be
allocated before or after vehicle dispatch. Although most previous
researchers have considered pre-dispatch guideway allocation approaches,
commonly referred to as "preprogrammed" or "reservation" schemes, this
author felt that such schemes, unless extensively modified, would be
impractical, and hence post-disposition systems should be developed
as it appears that these:

(1) Are more fault tolerant,
(2) Are better able to handle exiting backups,
(3) Can use the guideway for "dynamic storage",
(4) Have smaller entrance storage requirements,
(5) Are less vulnerable to shutdowns, and
(6) Do not require rigid time synchronization over
a whole guideway network.

As on-guideway longitudinal maneuvering is required in a post­
dispatch guideway allocation system, the following disadvantages
may occur with the latter:

1. Additional propulsion power would be consumed by maneuvering operations; and

2. Passenger discomfort could result, if maneuvering were excessive.

In the research reported here, one general approach to macroscopic control (termed dynamic scheduling) was developed. This approach was described in terms of a generalized dual-mode network model, and the controller architecture that was considered incorporates both a central coordinator and local interchange controllers. In practice, the control tasks concerned with individual vehicles (e.g., scheduling and maneuvering) would be handled at the local level, and those concerned with vehicles in the aggregate (e.g., "lane flow" distribution and "entrance rate" control) would be handled at the central level. The use of a combination of path reservations through interchanges (as each one is approached) and maneuvering-space reservations (for the "next" interchange) are the sine qua non of the dynamic scheduling scheme.

The flow potential of various interchange geometrics were investigated, and the directional type interchange was selected for further study. The approach lanes to such interchanges were designated as vehicle (longitudinal) maneuvering areas, and the various types of maneuvering trajectories required in a dynamic scheduling macroscopic control scheme were specified. One important maneuver that was specified is the "storage maneuver" which would allow a
limited number of vehicles to be delayed indefinitely on the approach to an interchange.

Both simple analytical queuing models and simulation models of interchange operation were developed. Analytical results obtained from the former were compared with empirical results obtained from the latter. Both clearly indicated that the maximum (saturation) flow rate through an interchange depends on the demand pattern on the same, and with random arrivals a limit of 0.75 vehicles/slot-time/lane was found for the balanced-demand case. Unbalanced demands resulted in higher obtainable flows. Interchange "cluster" configurations (which are physically expanded directional "interchanges" that have considerable distance prior to each merge area for longitudinal maneuvering) were modeled and found to have maximum per-lane flow densities approaching 1.00 vehicles/slot-time as pre-merge maneuvering distances were increased. This resulted from the reduction in maneuvering required on the approach lanes to the cluster, and hence, in greater achievable flows as a consequence of reduced interactions among maneuvered vehicles.

Corridor networks were defined as a series of entrance-exit facilities linked by two-way guideway sections, and models of various configurations were used to study the following:

1. Entrance-exit interchange (configuration) selection;
2. Lane-changing on multiple-laned guideway segments; and
(3) Entrance rejection under heavy demands.

The use of separate entrance (vehicle) storages for each direction of travel was found to result in higher obtainable flow rates and smaller vehicle delays. Also, dynamic scheduling procedures were found to be applicable to multiple-laned configurations and of distinct importance was a demonstration of the effectiveness of a simple swerve-maneuver policy wherein each lane spanning the distance between entrance-exit interchanges would be dichotomized into longitudinal- and lateral-maneuvering areas — longitudinal maneuvering being preparatory to lateral maneuvering. For a corridor network utilizing two lanes in each direction, the entrance rejection rate was found to be one-half that for a corresponding single-laned network (subject to one-half the demand). The resulting more than proportionate improvement in system capacity with such multiple-lane usage is attributed to a reduction in maneuvering interactions.

Simulation models were constructed for two different bidirectional-loop configurations, and for one of these, delay statistics were obtained for various demand levels, loop sizes (number of facilities on the loop), and trip lengths. Both the expected travel time and its dispersion increased with the demand level and with the number of facilities on the loop, and travelers taking long trips incurred more maneuvering delay than those on short trips. One troublesome condition with guideway loops was "network shutdown" which occurred when each interchange controller could not obtain maneuvering space (for vehicles it served) on downstream approaches
to following interchanges, and hence each interchange controller was forced to bring all vehicles on its approach lanes to a stop. Entrance metering was suggested as a procedure for reducing the probability of a shutdown, and should such a "deadlock" occur it was assumed that a limited number of vehicles could be required to exit from the system (and then reenter if they had not reached their destination).

Finally, the urban part of an intercity, dual-mode, automated transportation system (which would serve both as an interface with conventional city freeways and streets, and as an intra-city transportation mode) was modeled. The model included twelve interchange submodels and ten entrance-exit points. Two general classes of routing policies were discussed in terms of this network: demand responsive and demand independent. The latter was employed in the simulation model because of its simplicity. The average and standard deviation of delay were found to increase with both the demand level on the network and the number of interchanges a vehicle traversed on its trip.

B. Final Conclusions and Further Research

Probably the most important result of this study is the demonstration that viable deferred-disposition (post-dispatch guideway allocation) macroscopic control algorithms do exist for efficiently scheduling and routine vehicles over a rather general type of automated vehicle network. Previous researchers [30] had devised deferred-disposition approaches, but these were limited to a very restrictive class of networks which did not include conventional
highway interchanges. This is an unfortunate omission as a major constraint on allowable flow rates can be posed by the latter per the research reported here.

The following conclusions are based on the application of the dynamic scheduling algorithm to the various networks studied in this dissertation, and can probably be extrapolated to many other network configurations.

(1) A relatively high level of service appears to be obtainable. (The level achieved would be heavily dependent on the slot-time (T) with, for example, a flow rate of some 5800 veh/lane/hr corresponding to T = 0.5 sec).

(2) Flow rates through interchanges will in general be limited by the interactions among vehicles maneuvered on the approach lanes.

(3) The use of pre-merge maneuvering (maneuvering vehicles on the ramps of a directional interchange) increases achievable interchange flow levels. (A 33% increase resulted for the balanced demand case.)

(4) Both the expected total longitudinal maneuvering delay and its dispersion increase with both the level of demand on the system and the number of interchanges to be traversed by a traveler.

(5) Multiple-laned usage results in more than a proportionate increase in "system capacity" because there is, in general, reduction in maneuvering interactions.
(6) Flow control policies (such as entrance metering and routing procedures) should probably be designed to regulate flow levels so that system operation is at some level below "saturation". (In this way excessive vehicle maneuvering would not be required, and hence passenger discomfort would be avoided and propulsion energy and economics in usage would result.

(7) Based on the egress flow-limitation experiments, it appears that blockages would effectively be handled with a dynamic scheduling approach.

(8) Both entrance and exit storage areas are necessary; however, the required capacities of same are probably considerably smaller than those for pre-dispatch guideway allocation systems.

(9) The dynamic scheduling approach is amenable to both re-routing and re-scheduling vehicles around a blocked lane and restarting stopped vehicles after a blockage has been corrected.

As this research was only an initial step into the development of viable deferred-disposition macroscopic control algorithms, many fundamental concepts are yet to be discovered and many refinements must be made. Included in these required refinements are:

(1) The integration of various flow control policies into the dynamic-scheduling technique (e.g., entrance-
flow metering, dynamic-routing, and forced-exiting procedures);

(2) The evaluation and specification of the operating procedures to be used when various failure conditions (such as a lane blockage) occur;

(3) The development of more sophisticated maneuvering-space reservation procedures;

(4) The development of more detailed models of maneuvering section operation so that relationships among maneuvered vehicles can be accessed;

(5) The construction of aggregated-vehicle operational models so that larger networks can more easily be studied;

(6) The specification of microscopic control, communications, and computer requirements; and

(7) Perhaps a generalized network study so that network attributes, independent of a given network, can be discerned and appropriate conclusions concerning automated ground transport made.
APPENDIX I

THE DEFERRED DISPOSITION
TRANSPORT SIMULATOR (DDTS)
USER'S GUIDE

A. Introduction

This appendix provides information pertaining to the use of the Deferred - Disposition Transport Simulator (DDTS), while Appendix II provides a description of the logical structure of DDTS.

DDTS was developed explicitly for performing the simulation experiments described in this dissertation. DDTS is a system of programs written for use on a Digital Equipment Corporation PDP-10 computer (operating in a time-sharing monitor environment). The system was programmed by both the author and Bill Warner (who was responsible for the development of the BUILD Module of DDTS). The testing of DDTS was conducted on the Ohio State University 48-K PDP-10 installation (which is used by OSU's Computer and Information Department for graduate research). All routines are coded in either FORTRAN or MACRO (the PDP-10 assembler language). In general, assembler coding is used to perform those functions which are beyond the capabilities of PDP-10 FORTRAN or which could not be efficiently handled by same.

The DDTS System consists of ten functional modules. Only a single module is ever resident in core at any given time to reduce
the core requirements of DDTS. A copy of each module is retained on a mass storage device such as a high speed disk unit, and the control of module residence is one of the tasks of an executive module of DDTS called the SIMD module. Information flow between modules is accomplished through a common area of core and/or through data files on the mass storage device.

DDTS was designed for rapid construction of any model composed of the basic model elements described in Chapter IV. However, since it would require an extensive amount of effort for a user to specify the interconnection pattern for the hundreds of basic model elements that compose a moderate sized network, a special subnetwork model-building-block capability has been included. The user need only specify the interconnection of subnetworks (these represent interchanges, etc.) and then DDTS automatically expands the specified subnetworks into the required collections of basic model elements. For example, a "4W" subnetwork specification, i.e., a 4-way interchange model, generates 36 basic elements. These elements are then interconnected with those of neighboring interchanges, and the required inter-element information flow is set up (e.g., SG-to-SG feedback).

A DDTS user can set up several experiments, can specify the order in which they are to be run, and then enter the appropriate commands so that DDTS automatically executes the required simulations (without further user intervention). However, DDTS has special

\[1\text{For example, the simple urban network model developed in Chapter VII contains 334 basic model elements.}\]
"simulation-time" procedures which give the user the option of both influencing and observing the operation of a simulation which is in progress. This is accomplished by programming the user's teletype so that he can interrupt a simulation which is in progress. The simulation is temporarily suspended to process the user's commands.

DDTS automatically collects data from each simulation. One of the modules of DDTS is then used to process this data and produce a report which is stored on a mass storage device (for later transfer to a lineprinter or display device).

Fig. 86—Control transfer among modules.
# SIMU-Level Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>Set command input device and file.</td>
</tr>
<tr>
<td>PAUSE</td>
<td>Pause for continue request.</td>
</tr>
<tr>
<td>SETDO</td>
<td>Set DO-I/IIIMU Count.</td>
</tr>
<tr>
<td>DO</td>
<td>Start of DO-THRU loop.</td>
</tr>
<tr>
<td>THRU</td>
<td>End of DO-THRU loop.</td>
</tr>
<tr>
<td>JUMP</td>
<td>Jump to HERE command.</td>
</tr>
<tr>
<td>HERE</td>
<td>Take next command.</td>
</tr>
<tr>
<td>EXIT</td>
<td>Exit to PDP-10 Monitor.</td>
</tr>
<tr>
<td>SET</td>
<td>Set parameters.</td>
</tr>
<tr>
<td>SETUP</td>
<td>Chain in routine to assemble Common Core for compiled simulation.</td>
</tr>
<tr>
<td>INIT</td>
<td>Chain in routine to initialize state of simulator.</td>
</tr>
<tr>
<td>RUN</td>
<td>Chain in routine to perform simulation algorithm.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Save state of simulator.</td>
</tr>
<tr>
<td>GET</td>
<td>Get saved state of simulator.</td>
</tr>
<tr>
<td>DUMP</td>
<td>Prepare a formatted dump of simulation state.</td>
</tr>
<tr>
<td>OPEN</td>
<td>Open words of Common Core for observation or alteration.</td>
</tr>
<tr>
<td>STATS</td>
<td>Chain in routine to calculate statistics and prepare plots.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESTM</td>
<td>Run cure requirement estimating program.</td>
</tr>
<tr>
<td>BUILD</td>
<td>Run network compiler.</td>
</tr>
<tr>
<td>QUEUE</td>
<td>Branch to routine which sets up queue of compiled simulations.</td>
</tr>
<tr>
<td>NEXT</td>
<td>Get next compiled simulation.</td>
</tr>
<tr>
<td>RENM</td>
<td>Rename STATS output file to a unique name.</td>
</tr>
<tr>
<td>RSEED</td>
<td>Reset random number generator seed.</td>
</tr>
<tr>
<td>SNAP</td>
<td>Increment simulation max-time.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TECO</td>
<td>Run Teco Editor.</td>
</tr>
<tr>
<td>CTEC</td>
<td>Execute file CTEC.TEC as a TECO command file then return to simulator.</td>
</tr>
<tr>
<td>PROG</td>
<td>Run requested program from System or User's area.</td>
</tr>
<tr>
<td>B</td>
<td>BUILD</td>
</tr>
<tr>
<td>S</td>
<td>SETUP</td>
</tr>
<tr>
<td>SI</td>
<td>SETUP, INIT</td>
</tr>
<tr>
<td>SIR</td>
<td>SETUP, INIT, RUN</td>
</tr>
<tr>
<td>SIRS</td>
<td>SETUP, INIT, RUN, STATS</td>
</tr>
</tbody>
</table>

Fig. 87—The complete set of SIMU-level commands.
B. DDTS Module Control Transfer

The DDTS System consists of 10 modules as shown in Fig. 86. The usual way of accessing DDTS from the PDP-10 monitor command level is via a system "RUN" command (see the DEC System-10 User's Handbook). The command:

```
\$C
.RUN DSK:SIMU
```

results in the loading of the executive module of DDTS. This module indicates that it is waiting for SIMU-level commands by typing an "@" on the user's teletype. The user must respond with commands from the set defined in Fig. 87. The SIMU-level commands which result in control being transferred to other modules are also shown in Fig. 86.

When more than one command is entered, each must be separated by a comma. For example

```
@SETUP,INIT,RUN,STATS
```

is a command string which operates the four indicated modules in succession. However, the single "macrocommand":

```
@SIRS
```

serves the same purpose.

Other modules besides SIMU may be directly accessed from the PDP-10 monitor level as depicted in Fig. 88. Note that most of these return control to the SIMU Module upon completion. The user can return to the monitor level from any of the modules via the "$C" command (as illustrated in Fig. 88). A sequence of control transfers among DDTS
modules is illustrated in Fig. 89. Note that all transfers occur via the executive module of DDTS (Module SIMU) or via the PDP-10 Monitor.

Most of the modules have been designed to make use of the monitor system command "REEN" as illustrated in Fig. 90. The reentry points are defined by the Module Flow Charts (shown in Appendix II),
Fig. 89—An example of a sequence of control transfers among modules.

and the use of the "REEN feature" is discussed in the description of the Routine REEN (see Appendix II).
C. Conversation with DDTS

Table 1 illustrates a typical user session with DDTS. The teletype conversation with DDTS is shown on the left and explanatory comments are on the right. Note that all user keyboard inputs are underlined to distinguish them from DDTS output. The operations performed in this example are as follows:
(1) The user logs into the PDP-10 timesharing system;

(2) He accesses the ESTM Module of DDTS to estimate the core requirements of the urban network model to be constructed (see Fig. 68);

(3) The BUILD Module is used to construct the urban network model, and to generate the required O/D Matrices;

(4) The TECO Module is used to generate a title file;

(5) The SIMU-level "QUEUE" and "NEXT" commands are used to select files for DDTS processing;

(6) The SETUP Module is used to assemble data (produced by BUILD) into the required areas of COMMON Core;

(7) The INIT Module is used to initialize the state of the model with no vehicles on the modelled guideways;

(8) The RUN Module is used to perform the actual simulation;

(9) The STATS Module is used to prepare a statistical report;

(10) The system program PIP is used to obtain a listing of the report;

(11) The SIMU-level command "RENM" is used to save the report results in a file having a unique name; and

(12) Finally the user logs off the PDP-10 Computer System.
### Table 1

**Teletype Conversation with DDTS**

<table>
<thead>
<tr>
<th>Teletype Conversation with DDTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish communication with the monitor. Begin the login procedure and type in your identification. The job number assigned. Your password (It is not printed).</td>
</tr>
<tr>
<td>Load the ESTM Module core image and branch to starting address.</td>
</tr>
<tr>
<td>Enter the requested element parameters.</td>
</tr>
<tr>
<td>Enter subnetwork types and count of each for the network to be constructed.</td>
</tr>
<tr>
<td>A carriage return asks for a calculation of the number of element groups and an estimate of required N-array space. Note that these are within DDTS limits (ID&lt;160, IA&lt;2000) so network can be built.</td>
</tr>
<tr>
<td>The required number of various element group types can be obtained.</td>
</tr>
<tr>
<td>An &quot;S&quot; asks for the SIMD Module.</td>
</tr>
<tr>
<td>The SIMD Module is resident.</td>
</tr>
<tr>
<td>This is teletype 24-hour time-log information. An &quot;S&quot; means DDTS is waiting for a SIMD-level command. The BUILD Module is requested. The &quot;W&quot; means BUILD is waiting for an option. The subnetwork (SR) option is requested. Enter a carriage return (do not wish to save an intermediate file produced by BUILD). The teletype (TTY) is selected as the input device. A carriage return defaults to TTY also.</td>
</tr>
<tr>
<td>Enter requested parameters (free format).</td>
</tr>
<tr>
<td>Do not wish to change above parameter entries.</td>
</tr>
<tr>
<td>&quot;11&quot; means BUILD waiting for first subnetwork type A &quot;ON&quot; is selected. The node number is 1.</td>
</tr>
<tr>
<td>The 55-element ID is an &quot;A&quot;.</td>
</tr>
<tr>
<td>BUILD waits for the second subnetwork to be included in the network. Another &quot;ON&quot; is selected.</td>
</tr>
<tr>
<td>Continue in a similar manner for the remaining eight ON's.</td>
</tr>
<tr>
<td>A &quot;3W&quot; interchange subnetwork is selected.</td>
</tr>
<tr>
<td>The node connection and routing designator information is inputted. The node numbers must be entered in &quot;12&quot; format.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IC:</th>
<th>LOC 368.314</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job</td>
<td>PASSWORD:</td>
</tr>
<tr>
<td>RUN DEST:</td>
<td></td>
</tr>
<tr>
<td>ISDN:</td>
<td>7</td>
</tr>
<tr>
<td>C3 CAP:</td>
<td>156</td>
</tr>
<tr>
<td>SG CAP:</td>
<td>256</td>
</tr>
<tr>
<td>-SV 6</td>
<td></td>
</tr>
<tr>
<td>-SF 6</td>
<td></td>
</tr>
<tr>
<td>-SN 16</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>ID=123 IA=18458</td>
<td></td>
</tr>
<tr>
<td>-12</td>
<td></td>
</tr>
<tr>
<td>-55</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>-LK</td>
<td></td>
</tr>
<tr>
<td>-27</td>
<td></td>
</tr>
<tr>
<td>/SIMO/</td>
<td></td>
</tr>
<tr>
<td>/RDAS/</td>
<td></td>
</tr>
<tr>
<td>/BUILD/</td>
<td></td>
</tr>
<tr>
<td>SAVE 7.2</td>
<td></td>
</tr>
<tr>
<td>SUBWV: INPUT</td>
<td></td>
</tr>
<tr>
<td>DEV:FILE = TTY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTRCH LINK LTH, C3 LTH, CAP:</th>
<th>7, 168, 156</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG CAP:</td>
<td>1256</td>
</tr>
<tr>
<td>SG THRESHOLD:</td>
<td>54</td>
</tr>
<tr>
<td>C3 THRESHOLD:</td>
<td>156</td>
</tr>
<tr>
<td>CHANGE12</td>
<td></td>
</tr>
<tr>
<td>NODE:91</td>
<td></td>
</tr>
<tr>
<td>ID 1A</td>
<td></td>
</tr>
<tr>
<td>21CN</td>
<td></td>
</tr>
<tr>
<td>NODE: R</td>
<td></td>
</tr>
<tr>
<td>ID 10</td>
<td></td>
</tr>
<tr>
<td>31 END</td>
<td>...</td>
</tr>
</tbody>
</table>

1113W |
#115G |
#15ML |
#125C02 |
Table 1 (Cont)

Teletype Conversation with DDTS

<table>
<thead>
<tr>
<th>122.4K</th>
<th>13 O.C. J</th>
<th>17 D.E. F</th>
<th>#12-44-AGHI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>13) O.C. J,</td>
</tr>
<tr>
<td>17.4K</td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>#1. A</td>
<td></td>
<td></td>
<td>#11 SLJ</td>
</tr>
<tr>
<td>#12 FGH</td>
<td></td>
<td></td>
<td>#18 CEF</td>
</tr>
<tr>
<td>#18 AE.</td>
<td></td>
<td></td>
<td>#17 AE.</td>
</tr>
<tr>
<td>233LT</td>
<td></td>
<td></td>
<td>#16 AE.</td>
</tr>
<tr>
<td></td>
<td>1) CN A 1</td>
<td>8 8 8 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2) CN B 2</td>
<td>8 8 8 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3) CN C 3</td>
<td>8 8 8 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22) 4U</td>
<td>9 26 15 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23VR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FOLLOWING NODES SINGLY DEFINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;NODE&gt;&gt;</td>
</tr>
<tr>
<td>CHANGE &gt;&gt;</td>
</tr>
<tr>
<td>NAME = URBAN</td>
</tr>
<tr>
<td>=CD</td>
</tr>
<tr>
<td>NO OF STATIONS? : 18</td>
</tr>
<tr>
<td>MODE = 12</td>
</tr>
<tr>
<td>ENTRY = 88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATRIX 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D E F G H I J</td>
</tr>
<tr>
<td>A 80 80 80 80 80 80 80 80 80 80</td>
</tr>
<tr>
<td>B 80 80 80 80 80 80 80 80 80 80</td>
</tr>
<tr>
<td>C 80 80 80 80 80 80 80 80 80 80</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>J</td>
</tr>
</tbody>
</table>

| CHANGE >> |
| OUTPUT CNTX DEV: FILE = . |
| =CD |
| MODE = 14 |
| MODE = 12 |
| ENTRY = 11 |

<table>
<thead>
<tr>
<th>MATRIX 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C ...</td>
</tr>
<tr>
<td>A 8 1 1 ...</td>
</tr>
<tr>
<td>B 1 8 1 ...</td>
</tr>
<tr>
<td>C 1 1 1 ...</td>
</tr>
<tr>
<td>J</td>
</tr>
</tbody>
</table>

A second "3W" interchange is selected.

Note the great freedom in entering routing designators (as illustrated here).

Continue in a similar manner for the remaining four 3W's.

A "4U" interchange is selected.

The 4 nodes and associated routing designators are inputted.

Continue in a similar manner for the remaining five 4U's.

A listing of subnetworks and their connection nodes is requested.

The "VR" subcommand asks BUILD to compile the network.

No nodes were left open.

No changes are desired.

The name URBAN is selected for output files from BUILD.

The "4U" means BUILD is waiting for another option.

The "00" option is selected.

There are 10 20-elements (entrance-exit points).

Node 2 means a balanced demand.

80/1000 veh/slot-time.

The matrix you just constructed is as shown.

No changes necessary.

Output to default file (CNTX.DAT).

The "00" option is again selected.

Node 4 means filename CNTX.DAT is used.

Node is again requested (entry node).

Entry is 1 to select function #1.

The CNTX is as shown.
Table 1 (Cont)

Teletype Conversation with DDTS

<table>
<thead>
<tr>
<th>CHANLET</th>
<th>OUTPUT FMTX DEV=FILE = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>*EN</td>
<td></td>
</tr>
<tr>
<td>NUMBER OF FUNCTIONS: 1</td>
<td></td>
</tr>
<tr>
<td>FUNCTION 1: # POINTS: 2</td>
<td></td>
</tr>
<tr>
<td>ENTER FUNCTIONS: (0,0.0), (500,1.), (1000, 50)</td>
<td></td>
</tr>
<tr>
<td>CHANGE FUNCTION # ?</td>
<td>1</td>
</tr>
</tbody>
</table>

Done

/Simu/

/88121/

/TECO

TECO<<

*/1=SLK: URBAN NETWORK

/TITLE=DAT ( ) PWEY ( ) "G" ( ) |

*/Simu/

/88122/

/SDTiseum,NEXT,SIN,COND,STATS,PR

CND LIST -- SET QUEUE NEXT SETUP INIT

RUN DUMP STATS PROG

TIME, SNAP, SEED: 7: 15000

QUEUE<<

/FMTX|

/FMTX:|

/FUNCT:|

/NETWORK:|

>>

NEXT << *//FMTX/FUNCT/FUNCT/URBAN/>> |

SETUP<< >>

INIT << EMPTY NETWORK/ |

RUN << |

/I/TIME: 300 |

/I/TIME: 490 |

/I/TIME: 500 |

RUNTIME: 509 SEC |

STATS<< >>

PROG -- DEFINE =SYS: PIP |

/LETTERS=DAT, RAC, RP | |

/Run GEM1SIM |

/Simu/

/881/ |

/KEEN |

RUN << #& OUT FILE: 21508.RUN >> |

/*G |

/*G88 |

| Don't wish to change. |

/Output to FMTX.RAT (default). |

Requests the function generating option of BUILD. |

Only 1 function to be specified. |

It contains 3 points (linear interpolated). |

The coordinates are. (Note free format. The parentheses and commas are ignored.) No changes needed. |

"G" means go back to SIMU Module. |

BUILD Module is being left behind. |

SIMO Module is in core. |

The teletype time log. |

TECO Module is requested. |

A title file (TITLE.DAT) is made using TECO commands. Control is returned to the |

SIMO Module by the TECO exit command "70". |

A whole string of SIMO-level commands is |

imputed. |

The command list is as shown. Note that the |

"SIN" macrocommand was expanded. |

This is a result of the "GET" command. |

Note SNAP and SEED default to no "snaps" and |

standard seed. |

The file queues are entered. Carriage |

returns selects default names. Note, the |

"QUEUE" and "NEXT" commands are not really |

necessary here. However, the "QUEUE" |

command does check to see that all |

files are on disk. |

These are the selected files. |

SETUP Module operating. |

EXIT Module operating. |

REW Module starts operation. |

REW-time commands are entered. |

The simulation is terminated early by the |

"I" command. |

The computer run-time was 889 sec. |

The STATS Module is operated. |

The PIP program is accessed. |

Simulation results are printed on the line- |

printer. Control is transferred to the |

PDF-ID Monitor. |

The SIMO Module is loaded. |

The SIMO-level command "REW" is used to |

obtain a unique name for the results file. |

Control is transferred to the Monitor. |

The "kill job" routine is called so that |

the user can log off the PDF-ID System. |
Table 2

<table>
<thead>
<tr>
<th>SIMU Module Messages</th>
<th>Meaning and Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SIMU/</td>
<td>A fresh copy of SIMU has just been loaded into core. Common core is cleared.</td>
</tr>
<tr>
<td>/hh:mm/</td>
<td>24-hour time log. (See LOG Routine).</td>
</tr>
<tr>
<td>&lt;SIMD&gt; CMD SRC &lt;DEV:FILE —</td>
<td>DOTS needs to know what device and file it can find the list of SIMU-level commands. A carriage return results in the TTY being selected.</td>
</tr>
<tr>
<td>DEVICE &lt;dev&gt; NOT LEGAL</td>
<td>Try again. Format is &quot;dev: name&quot; and extension DAT is assumed.</td>
</tr>
<tr>
<td>INPUTTING (name.DAT, LOGO = nm)</td>
<td>DOTS is reading in the SIMU-level command file.</td>
</tr>
<tr>
<td>0 _</td>
<td>DOTS is waiting for one line of SIMU-level commands. A single command or a string of commands delimited by commas may be entered.</td>
</tr>
<tr>
<td>CMD LIST -- command 1, command 2, ...</td>
<td>This is a verification of the stored command list. (This is published only if there are more than six expanded SIMU-level commands.)</td>
</tr>
<tr>
<td>&lt;SIMD&gt;</td>
<td>This is the result of a &quot;PASS&quot; command. Type &quot;YES&quot; or &quot;NO&quot;.</td>
</tr>
<tr>
<td>(command) IS NEXT...CONTINUE ? : ___</td>
<td>A carriage return is equivalent to a &quot;YES&quot;.</td>
</tr>
<tr>
<td>&lt;SIMD&gt;</td>
<td>This is the result of a &quot;SETDU&quot; command. The value entered controls looping through commands between the &quot;DO&quot; and the &quot;END&quot; commands.</td>
</tr>
<tr>
<td>SET DO THRU COUNT TO ? : ___</td>
<td>Reenter the SIMU-level command list using correct format and legal commands.</td>
</tr>
<tr>
<td>ILL CMD &lt;command&gt;</td>
<td>This is the result of a &quot;SET&quot; command. Enter TIME and SNAP in slot-times. A SEED of zero defaults to standard RAND Routine seed. A free format is used for all three parameters.</td>
</tr>
<tr>
<td>TIME, SNAP, SEED ? : ___</td>
<td>This is the result of a &quot;PROG&quot; command. Valid responses are, for example, 975:PIF or 050:ESIM .</td>
</tr>
<tr>
<td>PROG -- DEV:NAME =</td>
<td>&quot;SAVE&quot; command is executed. Entrance to the SAVE Routine results in &quot;&lt;&lt; &quot; being typed and exit from routine types &quot;&gt;&gt;&quot;.</td>
</tr>
<tr>
<td>SAVE &lt;&lt; &gt;&gt;</td>
<td>&quot;GET&quot; command is executed.</td>
</tr>
<tr>
<td>GET &lt;&lt; &gt;&gt;</td>
<td>Result of &quot;QUEUE&quot; command. Enter filename strings. Use commas to delimit. A carriage return results in default names.</td>
</tr>
<tr>
<td>QUEUE &lt;&lt; file type: ___</td>
<td>Some files specified to QUEUE were not found on disk.</td>
</tr>
<tr>
<td>MISSING -- file1/file2/.../fileN/</td>
<td>All files specified to QUEUE were found.</td>
</tr>
<tr>
<td>/ALL FOUND /</td>
<td>Files being renamed by &quot;NEXT&quot; command.</td>
</tr>
<tr>
<td>NEXT &lt;&lt; file1/file2/file3/file4 &gt;&gt;</td>
<td>Filename used by a &quot;RENM&quot; command when renaming GOODS.DAT .</td>
</tr>
<tr>
<td>RENM &lt;&lt; L&amp;OUT FILE: name.RIN &gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Table 2 (Cont)

DDTS Messages and the Required Responses

<table>
<thead>
<tr>
<th>ESTM Module Messages</th>
<th>Meaning and Action required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LK LGR:__</td>
<td>Enter length of interchange LK elements.</td>
</tr>
<tr>
<td>OZ CAP:__</td>
<td>Enter OZ-element capacity.</td>
</tr>
<tr>
<td>SG CAP:__</td>
<td>Enter SG-element capacity.</td>
</tr>
<tr>
<td>__</td>
<td>Enter subcommand to ESTM. XX,mm (where XX is C, E, J, or W and mm is the number of such subnetworks), or enter XX (where XX is G, O, S, L, M, or S to get a group count). Other subcommands are listed in the ESTM Routine description.</td>
</tr>
<tr>
<td>ID = mmm 1A= fffff</td>
<td>This is the response to a &quot;carriage return&quot; subcommand. It is the estimate of the total number of groups and the M-array space required. Current DDTS limits are ID &lt; 160, fffff &lt; 20000.</td>
</tr>
<tr>
<td>__</td>
<td>This is the response to a group-count subcommand.</td>
</tr>
<tr>
<td>__</td>
<td>An illegal subcommand. Try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SETUP Module Messages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SETUP &lt;&lt;</td>
<td>The SETUP Module is resident and operating.</td>
</tr>
<tr>
<td></td>
<td>The SETUP Module is chaining in the SIMC Module.</td>
</tr>
<tr>
<td></td>
<td>The sizes of the O/D Matrices (ODMTX and FNTX) do not agree with the number of ES-elements specified by the file SETUPI.DAT.</td>
</tr>
<tr>
<td></td>
<td>The network is larger than can be handled by the current version of DDTS. The number of element-groups must be ≤ 160.</td>
</tr>
<tr>
<td></td>
<td>The M-array storage limit of DDTS is 20000. Either reduce size of the network by parameter and/or configuration changes or increase COMMON dimensions and re-compile DDTS.</td>
</tr>
<tr>
<td></td>
<td>Use BUILD Module to generate required files.</td>
</tr>
<tr>
<td></td>
<td>Reduce appropriate parameters.</td>
</tr>
<tr>
<td></td>
<td>Reduce appropriate parameters.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INIT Module Messages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT &lt;&lt; EMPTY NETWORK &gt;&gt;</td>
<td>An &quot;INIT&quot; command is executed. Left and right angle brackets correspond to beginning and end of module execution.</td>
</tr>
</tbody>
</table>
## Table 2 (Cont)

### DDT S Messages and the Required Responses

<table>
<thead>
<tr>
<th>RUN Module Messages</th>
<th>Meaning and Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN &lt;&lt;</td>
<td>Beginning of RUN Module.</td>
</tr>
<tr>
<td></td>
<td>End of RUN Module.</td>
</tr>
<tr>
<td>N-BIT ENABLE ? : ___</td>
<td>This is of historic importance (should be removed from DDT S). Type &quot;carriage return&quot; to continue simulation.</td>
</tr>
<tr>
<td>RUNTIM = mnnn SEC</td>
<td>Run-time for the RUN module. This is not simulation time.</td>
</tr>
<tr>
<td>BAD TABLE AT N(mm) :: LOC a,T = 0</td>
<td>DDT S table is bad. Did you forget to do a SETUP and INIT? See TEST Routine description.</td>
</tr>
<tr>
<td>ERR AT T = tttt</td>
<td>A simulation error of type &quot;a&quot; occurred at given time (tttt). Simulation is continuing.</td>
</tr>
<tr>
<td>/run cmd response</td>
<td>This is general form of the RUN-time command responses. E.g. / T PLO TS, TIME=mnnn .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATS &lt;&lt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>... DATA.DAT BAD ?</td>
</tr>
<tr>
<td>... EOF ON DATA.DAT ?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPEN, TECO, CT EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN &lt;&lt; }</td>
</tr>
<tr>
<td>TECO &lt;&lt; }</td>
</tr>
<tr>
<td>CT EC &lt;&lt; }</td>
</tr>
<tr>
<td>?m</td>
</tr>
<tr>
<td>?</td>
</tr>
<tr>
<td><strong>BUILD Module Messages</strong></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td><em>__</em></td>
</tr>
<tr>
<td>&lt;&lt;&lt;x&gt;&gt; IS AN ILLLEGAL OPTION:</td>
</tr>
<tr>
<td>LIST IS: 50, 00, 10, 70, ES, SB —</td>
</tr>
<tr>
<td>TRY AGAIN. ARRAY FULL</td>
</tr>
<tr>
<td>DORE</td>
</tr>
</tbody>
</table>

**Messages for ES option**

| **SAVE ?__**               | "Y" or "N" (historic) (enter carriage return) |
| **SUBMITX INPUT DEV:FILE =__** | Carriage return selects TTY. |
| **ENTER LINK LOTH, OZ LOW, CAP?__** | Enter all three values (free format). |
| **SG CAP?__**              | SC-element capacities (free format). |
| **SG THRESHOLD?__**        | SG-element threshold (free format). |
| **OZ THRESHOLD?__**        | OZ-element threshold — usually set equal to capacity (free format). |

| **CHANGE?__**              | Allows one to change above parameters. Answer "Y" or "N" (carriage-return defaults to "N"). |
| **__num__**                | Enter one of the following: 42, 34, ES, SC, 90, EA, SS, or LT. See SUBMITX routine description. ("num" is element number). |
| **__EXIT__**               | Enter a two-digit node number followed by destination designating letters. Letters may be adjacent or delimited by as many non-alphabetic characters as desired. E.g., "21ZADEF" or "36A tab-0.F...C" both enter same information. |
| **__EX STATION ID?__**     | Enter a single letter to designate EX-facility. |
| **__NODE:__**              | Node and designator for an ON subnetwork. (This simply connects up an SS element.) |
| **__ID:__**                | A single node can only be common to two subnetworks. Enter element again. |
| **NODE <<<x>> ALREADY DEFINED TWICE** | You repeated an alphabetic designator. No problem, but maybe you made a typing error? |
| **DEST <<<x>> MULI DEFINED IN ABOVE** | Two SS-element-creating-subnetworks (ON's and ES'as) have used the same alphabetic ID. |
| **<<<x>> ALREADY A STATION** | You skipped this letter in the alphabet (when defining ON's and ES'as). Re-specify network. |
| **STATION <<<x>> IS A NULL STATION** | Enter EX length for an SS subnetwork. |
| **LENGTH:__**              | |
Table 2 (Cont)

DDTS Messages and the Required Responses

<table>
<thead>
<tr>
<th>BUILD Module Messages (cont)</th>
<th>Meaning and Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOLLOWING NODES SINGLY DEFINED either a1, a2, ... or &lt;SOMETHING&gt;</td>
<td>This just lets the user know that some nodes are open. No problem, if user's intention.</td>
</tr>
<tr>
<td>CHANGE ? : ___</td>
<td>Answer &quot;Y&quot; if you wish to change any elements, otherwise &quot;carriage return&quot;.</td>
</tr>
<tr>
<td>ERR LOC ? :</td>
<td>Give element-number to be corrected.</td>
</tr>
<tr>
<td>ERROR MUST BE BEFORE CURRENT POSITION</td>
<td>Improper response to ERR LOC? : ___</td>
</tr>
<tr>
<td><em><strong>WARNING</strong></em> ARRAY N BECOMING FULL, SIZE = mnn</td>
<td>Limitations of BUILD Module exceed or about to be exceeded.</td>
</tr>
<tr>
<td><em><strong>ERROR</strong></em> ARRAY N OVERFILLED</td>
<td></td>
</tr>
<tr>
<td><em><strong>WARNING</strong></em> ARRAY SUM NEARLY FULL</td>
<td></td>
</tr>
<tr>
<td>NODE NUMBERS MUST BE LESS THAN 101</td>
<td></td>
</tr>
<tr>
<td>TOO MANY ENTRIES FOR ARRAY TIETAB IN SOTIE</td>
<td></td>
</tr>
<tr>
<td>TABLE ERROR</td>
<td></td>
</tr>
<tr>
<td>ARRAYS ARE FULL IN SOTIE2</td>
<td></td>
</tr>
<tr>
<td>NAME? : ___</td>
<td>See explanation of KNANE Routine.</td>
</tr>
<tr>
<td>&lt;(name)&gt; ALREADY IN USE</td>
<td>More than one node of an interchange has been assigned a common destination.</td>
</tr>
<tr>
<td>&lt;&lt;no&gt;&gt; HAS ILLEGAL ROUTE</td>
<td></td>
</tr>
<tr>
<td>Messages for OD option</td>
<td></td>
</tr>
<tr>
<td>NO OF STATIONS? : ___</td>
<td>Enter number of SS elements in network (free format).</td>
</tr>
<tr>
<td>NODE? : ___</td>
<td>Enter Matrix input node desired.</td>
</tr>
<tr>
<td>ENTRNT : ___</td>
<td>Enter a single matrix entry.</td>
</tr>
<tr>
<td>A B C ...</td>
<td>Enter matrix row-by-row (free format).</td>
</tr>
<tr>
<td>A ___ ___ ___</td>
<td>Matrix is listed for checking by user.</td>
</tr>
<tr>
<td>B ___ ___ ___</td>
<td>&quot;Y&quot; or &quot;N&quot; (carriage return equiv. to &quot;N&quot;).</td>
</tr>
<tr>
<td>MATRIX IS etc.</td>
<td>Answer in format X,Y (to designate row and col.).</td>
</tr>
<tr>
<td>CHANGE? ___</td>
<td>Enter correct matrix entry.</td>
</tr>
<tr>
<td>CHANGE MTX (? ,?) ___</td>
<td>Enter a legal row and col.</td>
</tr>
<tr>
<td>MTX(x,y)= ___</td>
<td></td>
</tr>
<tr>
<td>ERROR IN INPUT X AND/OR Y IS ILLEGAL</td>
<td></td>
</tr>
<tr>
<td>OUTPUT filename DEV: FILE &quot;_&quot;</td>
<td>A carriage return outputs matrix to &quot;filename&quot;. If another file is to be used specify DEV: FILE.</td>
</tr>
</tbody>
</table>
### Table 2 (Cont)

**DDTS Messages and the Required Responses**

<table>
<thead>
<tr>
<th><strong>BUILD Module Messages (cont)</strong></th>
<th><strong>Meaning</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER OF FUNCTIONS:</td>
<td>(free format).</td>
</tr>
<tr>
<td>FUNCTION no. II PAIRS:</td>
<td>Give no. of (x,y) pairs to be entered.</td>
</tr>
<tr>
<td>ENTER FUNCTION:</td>
<td>Enter all (x,y) pairs (free format).</td>
</tr>
<tr>
<td>CHANGE FUNCTION # ?:</td>
<td>Enter a function number if you wish changes. Otherwise a carriage return.</td>
</tr>
</tbody>
</table>
Table 2 (Cont)

<table>
<thead>
<tr>
<th>Module independent Messages</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE: ___</td>
<td>DDTS is waiting for a reentry point. (See description of RENK Routine.)</td>
</tr>
<tr>
<td>...NO GOOD...</td>
<td>An illegal reentry point was selected. Try again.</td>
</tr>
<tr>
<td>*** CORHDU FAILURE ***</td>
<td>Routine CORHDU is unable to obtain the core required for a given module. Some other job may be locked into core. Wait until core is available.</td>
</tr>
<tr>
<td>BYTE PACK ERR...PC = nnnnnn</td>
<td>This message is published by routines PKR2 and PKG4. Either element parameters are out of range, or user requested a RENK without a SETUP and UNIT.</td>
</tr>
<tr>
<td>I-60.</td>
<td>An error return to Module SIMU. A new SIMU-level command list is required.</td>
</tr>
<tr>
<td>INTEGER OVERFLOW PC= nnnnnn</td>
<td>These are Fortran Operating System detected errors. A DOTS limitation may have been exceeded, if these occur in any module except STATS. Occurrence of these messages in STATS is usually acceptable.</td>
</tr>
<tr>
<td>FLOATING OVERFLOW PC= nnnnn</td>
<td>This is the first line of a monitor detected error. These errors are always fatal. Such errors either represent: user exceeding limitations of DOTS, a bug in DOTS, or an error in the PDP-10 Monitor.</td>
</tr>
<tr>
<td>ERROR IN JOB n...</td>
<td>Occurs when several large core users are using the PDP-10 system. May be a result of a bad track on the disk. Try again.</td>
</tr>
<tr>
<td>? SNAP READ ERROR</td>
<td>Attempt to start a program after a fatal error has occurred, or attempt to RENK a module which has no RENK points (e.g., ESTM).</td>
</tr>
<tr>
<td>? NO START ADDRESS</td>
<td></td>
</tr>
</tbody>
</table>


Table 3

**RUN-time Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Wrap-Up Run get next SIMU command.</td>
</tr>
<tr>
<td>@</td>
<td>Wrap-Up Run go to SIMU Command-in mode</td>
</tr>
<tr>
<td>*</td>
<td>Take string of RUN commands</td>
</tr>
<tr>
<td>.</td>
<td>End RUN-command string, otherwise string (started by an *) is ended by exhausting all characters in TTY-in buffer</td>
</tr>
<tr>
<td>D</td>
<td>Do a SIMU DUMP command, then continue simulation</td>
</tr>
<tr>
<td>%</td>
<td>Type OZ flows in percent of slots used.</td>
</tr>
<tr>
<td>Z</td>
<td>Type No. of vehicles in OZ's.</td>
</tr>
<tr>
<td>S</td>
<td>Type No. of vehicles in SC's.</td>
</tr>
<tr>
<td>N</td>
<td>Type No. of Trips matrix.</td>
</tr>
<tr>
<td>↑</td>
<td>Change protection (flip-flop)</td>
</tr>
<tr>
<td>←</td>
<td>Change feedback (flip-flop)</td>
</tr>
<tr>
<td>T</td>
<td>Current simulation time.</td>
</tr>
<tr>
<td>TYPE</td>
<td>EXPLANATION</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>General</td>
</tr>
<tr>
<td>2</td>
<td>An SG is over-capacity.</td>
</tr>
<tr>
<td>3</td>
<td>An OZ is over-capacity.</td>
</tr>
<tr>
<td>4</td>
<td>An MG collision occurred.</td>
</tr>
<tr>
<td>5</td>
<td>A vehicle arrived at the wrong destination.</td>
</tr>
</tbody>
</table>
It is suggested that a user actually try the example shown in Table 1 for the purpose of becoming familiar with the DDTS System. There are many facets of conversation with DDTS that can only be learned with experience. Table 2 is a documentation of most of the messages that a user might encounter when using DDTS. The meaning of each message and the action required are given. The messages are grouped according to the module by which they are published. Module independent messages (e.g., those published by the PDP-10 Monitor) are listed in a special group of Table 2.

Table 3 lists the "RUN-time commands" which can be used while the RUN Module is operating, and Table 4 lists simulation errors.

D. DDTS Application Examples

The manner in which the urban model (see Fig. 68) was constructed and operated is illustrated by Table 1. Similar model specifying diagrams are shown in Figs. 91-95. Note that the LINE1 Model (shown in Fig. 43) is constructed in a straightforward manner using DDTS (see Fig. 91) whereas the LINE2 and LINE3 Models, constructed via the subnetwork-specifying-procedure of DDTS, result in many redundant basic elements being included in the models. The user could use the basic-element-specifying procedure of DDTS for generating more efficient models; however, using the subnetwork building blocks in the manner illustrated by Figs. 91-95 considerably simplifies the model specifying task. The form of the O/D matrices required for the "LINE Models" are shown in Figs. 96-97. Note that there are many zeroes in the matrices, due to redundant model elements.
Fig. 91—Specifications for the LINE1 Model.
Fig. 92—Specifications for the LINE2 Model.
Fig. 93—Specifications for the LINE3 Model.
Fig. 94—Specifications for the CLUSTER-3 Model.
Fig. 95—Specifications for the CLUSTER-4 Model.
Fig. 96—The required O/D matrix form for the LINE1 and LINE2 Models.
Fig. 97—The required O/D matrix form for the LINE3 Model.
APPENDIX II
THE DDTS INTERNAL LOGIC MANUAL

This appendix provides additional information pertaining to the logical structure of DDTS. Details include a presentation of the simulation algorithm, a set of module flow diagrams, and a description of each routine contained within the DDTS Routine Library.

A. The Simulation Algorithm

The simulation algorithm employed by DDTS is the procedure by which DDTS generates entities (vehicle-words), processes these entities, and then eventually destroys them. The generation of an entity represents the arrivals of vehicles to the entrance of the automated guideway system. The processing of these entities represents the movement of vehicles through the system, and the destruction of entities represents the departure of a vehicle from the automated network.

A vehicle-word is a 36-bit PDP-10 computer word coded according to the format depicted in Fig. 98. Note that each word consists of four bytes. The "destination byte" is an ASCII-coded alphabetic character which represents the destination point of the
entity. This information is used in routing, scheduling, and data collection procedures involving the vehicle entity. Similarly the "origin byte" is an alphabetic character representing the origin point of the entity. It is used in the data collection procedures (which calculate O/D matrix statistics). The "temporary data byte" is used for storing departure time information for a vehicle entity logically located within an OZ element. The "time byte" is used to record the slot-time at which a vehicle entity is generated. This information is used (by the sink portion of an SS element) in calculating vehicle delay statistics.

Simulation events are described as the movement of vehicle entities from one basic model element to another. To avoid event time-sequencing problems, special buffers termed "element-in" and
"element-out" buffers are utilized. These can be thought of as representing the input and output nodes, respectively, of the model elements. During each simulated slot-time a group of routines termed the "Phase 1 routines" move vehicle-words into the output-buffers associated with the model elements comprising the simulation. These entities are moved from memory locations logically associated with each of the model elements. The elements are logically tied together by a table which controls the transfer of entities (once each slot-time) from the output-buffers to the appropriate element input-buffers. A group of "Phase 2 routines" are called (after the table-transfer) to absorb the input-buffer entities and place them in memory locations logically associated with the various model elements. The complete algorithm performed by the RUN Module of DDTS is as follows:

1. Update time-varying demand matrix.

2. Perform Feedback Algorithm.
   a. Turn on all SG gates
   b. Turn off all SG gates which feed OZ sections that are at capacity threshold.
   c. Turn off all SG gates feeding SG elements that are at or above threshold.

3. Perform Phase 1 routines for each Element, moving a Vehicle-Word into each Element-Out-Buffer.

4. Transfer Vehicle-Words from Element-Out-Buffers to Element-In-Buffers (The Network Transfer Table specifies the element connections).

5. Perform Phase 2 Routines for each Element, absorbing a Vehicle-Word from each Element-In-Buffer.
(6) Output time-series data to a disk file (every 100 slot-times).

(7) Output simulation error conditions to the teletype. Such conditions are detected in the Phase 1 and Phase 2 routines.

(8) Increment slot-time.

(9) Output job status to a disk file (every 1000 slot-times).

(10) Execute RUN commands found in teletype input buffer (every 100 slot-times).

(11) Check for end-simulation condition (slot-time equals max-time). Go to (1) if condition not met.

B. DDTS logic

In the internal structure of DDTS, model elements are logically categorized into classes termed "element groups." An element group consists of either two or four elements, depending on the element type. (Elements of different types are never mixed.) The number of elements per group is determined by the limit of four input and four output vehicle-word buffers per group. Since each MG element has two input buffers, only two such elements can be packed into a group. (Also two of the output buffers in such a group are not used.) The number of elements per group is determined by the limit of four input and four output vehicle-word buffers per group. Since each MG element has two input buffers, only two such elements can be packed into a group. (Also two of the output buffers in such a group are not used.) Similarly, there are only two elements in a DG group. LK, SG, OZ, and SS elements are packed four to a group.
<table>
<thead>
<tr>
<th>ELEMENT TYPE</th>
<th>No. Char.</th>
<th>Mnemonic</th>
<th>ME(1D)</th>
<th>REGISTERS</th>
<th>L (1,ID)</th>
<th>L (2,ID)</th>
<th>L (3,ID)</th>
<th>L (4,ID)</th>
<th>L (5,ID)</th>
<th>L (6,ID)</th>
<th>L (7,ID)</th>
<th>L (8,ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK</td>
<td>1</td>
<td>LK</td>
<td>1</td>
<td>Length</td>
<td>Low Bound</td>
<td>Upper Bound</td>
<td>Pointer</td>
<td>--</td>
<td>--</td>
<td>Number of Vehicles (packed)</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>STORE/GATE</td>
<td>1</td>
<td>SG</td>
<td>3</td>
<td>Capacity, Threshold</td>
<td>Low Bound</td>
<td>Upper Bound</td>
<td>In Pointers (packed)</td>
<td>Out Pointers (packed)</td>
<td>Gate Logic (packed)</td>
<td>Number of Vehicles (packed)</td>
<td>Address of STORE TABLE (Low)</td>
<td></td>
</tr>
<tr>
<td>ORGANIZE</td>
<td>4</td>
<td>OZ</td>
<td>4</td>
<td>Length, Capacity, Threshold</td>
<td>Low Bound, No. OZ</td>
<td>Merge Red Point, Merge Red Length (word)</td>
<td>In Pointers (packed)</td>
<td>Out Pointers (packed)</td>
<td>Last asked time cont in L (8,ID)</td>
<td>Number of Vehicles (packed)</td>
<td>Last asked time (cont)</td>
<td></td>
</tr>
<tr>
<td>SOURCE/SINK</td>
<td>1</td>
<td>SS</td>
<td>5</td>
<td>Number of SS elements (Low)</td>
<td>Low Bound (Source Info) (Low)</td>
<td>Upper Bound (Source Info) (Low)</td>
<td>Low Bound (Sink Info) (Low)</td>
<td>Start of QNUMX (Low)</td>
<td>(No. SS)</td>
<td>Rejects</td>
<td>Rejects (cont)</td>
<td></td>
</tr>
<tr>
<td>MERGE</td>
<td>0</td>
<td>MG</td>
<td>6</td>
<td>No. Collisions</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>NODE INFO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVERGE</td>
<td>2</td>
<td>DG</td>
<td>7</td>
<td>Diverge table I</td>
<td>Diverge table II</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 99—The element-group register format.
since each element only requires a single input and a single output buffer. Element grouping is used to reduce core storage requirements and to increase efficiency. For example, all elements within a group must have the same parameter values, and hence several variables can be common to a group. Each group has a set of eight such variables termed the element-group registers. The format of these registers depends on the group type as illustrated in Fig. 99. Some of these registers are used to "point" to areas in DDTS's general storage area (termed the M-array). All vehicle-words "contained" within an element are stored in the M-array area associated with the element.

The logical structure of the DDTS modules are contained in the flow charts displayed in Fig. 100-106. These diagrams provide only a very macroscopic description of DDTS. More details are contained in the next section where the complete library of routines comprising DDTS are described.

C. The DDTS Library Routines

Each of the ten DDTS modules consist of one or more routines from a set of routines termed the DDTS Library. These relocatable routines are linked together using the PDP-10 Relocatable Loader, and a core image of each module is saved on a mass storage device.

Table 5 is an alphabetical listing of the 109 routines used in DDTS. Eleven of these routines are from the Fortran Library (LIB40). The next section of this chapter contains a description of each of these routines. (These are arranged in alphabetical order.)
Fig. 100—SIMU Module flow chart.
Fig. 101—ESTM Module flow chart.
Type "A" wait for option and decode

Sort elements according to type.
Sort elements according to characteristics.
Assign elements to groups.
Assign ID's to groups.
Count # of groups of each type.
Count # of SS elements.
Count # of elements in each group.

Make vehicle-word element-buffer assignments.

Make network buffer transfer table. Write file NETWK.DAT

Call SGTIE2
(produces file SGTIE.DAT)

Call RNNAME

Fig. 102—BUILD Module flow chart.
Fig. 103—SETUP Module flow chart.
Initialize Routine

Simu-time set to zero

M ARRAY initialized to empty-slot words

Element Buffers initialized to empty-slot words

Set so that SIMU asks for a new command string

Fig. 104--INIT Module flow chart.
1. Update time-varying demand matrix.

2. Perform Feedback Algorithm.
   a. Turn on all SC gates
   b. Turn off all SG gates which feed OZ sections that are at capacity threshold.
   c. Turn off all SG gates feeding SG elements that are at or above threshold.

3. Perform Phase 1 Routines for each Element, moving a Vehicle-Word into each Element-Out-Buffer.

4. Transfer Vehicle-Words from Element-Out-Buffers to Element-In-Buffers (The Network transfer Table specifies the element connections).

5. Perform Phase 2 Routines for each Element, absorbing a Vehicle-Word from each Element-In-Buffer.

6. Output time-series data to a disk file (every 100 slot-times).

7. Output simulation error conditions to the teletype. Such conditions are detected in the Phase 1 and Phase 2 routines.

8. Increment slot-time.

9. Output job status to a disk file (every 1000 slot-times).

10. Execute RUN commands found in teletype input buffer (every 100 slot-times).

Fig. 105—RUN Module flow chart.
Contract core to JOBFF

Define .REEN points

Initialize
- find date
- find time
- read TITLE file
- open output file (GOODS)
- write out title

Write out
O/D NO. TRIPS MATRIX
NO. OF REJECTS

Calculate O/D statistics
Flows, Mean and Std-Dev (Delay)

Write out
- Actual and generated density matrices
- MEAN and STD-DEV delay matrices
- MAX and MIN matrices

Calculate MAX/MEAN
Delay MEAN/MIN
Ratios

Write out Ratio Matrices

Plot interchange flows
(out-node of each OZ element)

Fig. 106—STATS Module flow chart.
The routines used in each of the DDTS modules are shown in Figs. 107-111. The nesting of routines is also shown. (Routines to the right are called by the ones on the left.) The utility routines referred to in these diagrams are any of the general input-output (e.g., IFILE) or general data packing-unpacking routines (e.g., PKUP).
Fig. 107—SIMU Module routine nesting.
Fig. 108—BUILD Module routine nesting.
Fig. 109—ESTM, SETUP, AND INIT Module routine nesting.
Fig. 110—RUN Module routine nesting.
Fig. 111--STATS Module routine nesting.
<table>
<thead>
<tr>
<th>DDTS</th>
<th>Routines</th>
<th>(Alphabetical Listing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>BELL</td>
<td>PK04</td>
</tr>
<tr>
<td>2.</td>
<td>BUFF</td>
<td>PKDATA</td>
</tr>
<tr>
<td>3.</td>
<td>BUILD</td>
<td>PKDEST</td>
</tr>
<tr>
<td>4.</td>
<td>BYTE</td>
<td>PKORIG</td>
</tr>
<tr>
<td>5.</td>
<td>CHAIN</td>
<td>PKTIME</td>
</tr>
<tr>
<td>6.</td>
<td>CHECK</td>
<td>PKUP</td>
</tr>
<tr>
<td>7.</td>
<td>COMMON (DAT.)</td>
<td>PLOTS</td>
</tr>
<tr>
<td>8.</td>
<td>C/RUO</td>
<td>POUT</td>
</tr>
<tr>
<td>9.</td>
<td>CTEC (MAIN.)</td>
<td>PPUT</td>
</tr>
<tr>
<td>10.</td>
<td>DATA</td>
<td>QUEUE</td>
</tr>
<tr>
<td>11.</td>
<td>DATE</td>
<td>RAN</td>
</tr>
<tr>
<td>12.</td>
<td>DELIM</td>
<td>REEN</td>
</tr>
<tr>
<td>13.</td>
<td>DEVFIL</td>
<td>REGS</td>
</tr>
<tr>
<td>14.</td>
<td>DG1</td>
<td>RELEASE</td>
</tr>
<tr>
<td>15.</td>
<td>DG2</td>
<td>RENAM</td>
</tr>
<tr>
<td>16.</td>
<td>DUMP</td>
<td>RENM</td>
</tr>
<tr>
<td>17.</td>
<td>DWORD</td>
<td>RESET</td>
</tr>
<tr>
<td>18.</td>
<td>ENTER</td>
<td>RING</td>
</tr>
<tr>
<td>19.</td>
<td>ENTR</td>
<td>RNAME</td>
</tr>
<tr>
<td>20.</td>
<td>ESTM (MAIN.)</td>
<td>RUN</td>
</tr>
<tr>
<td>21.</td>
<td>EXIT</td>
<td>RUNCMD</td>
</tr>
<tr>
<td>22.</td>
<td>EXPND</td>
<td>RUNTIM</td>
</tr>
<tr>
<td>23.</td>
<td>FN</td>
<td>RUNUO</td>
</tr>
<tr>
<td>24.</td>
<td>GATEON</td>
<td>SAVE</td>
</tr>
<tr>
<td>25.</td>
<td>GET</td>
<td>SAVRAN</td>
</tr>
<tr>
<td>26.</td>
<td>GRP</td>
<td>SETRAN</td>
</tr>
<tr>
<td>27.</td>
<td>IFILE</td>
<td>SETUP</td>
</tr>
<tr>
<td>28.</td>
<td>INIT</td>
<td>SG1</td>
</tr>
<tr>
<td>29.</td>
<td>IOACCP</td>
<td>SG2</td>
</tr>
<tr>
<td>30.</td>
<td>IONTR</td>
<td>SG1E2</td>
</tr>
<tr>
<td>31.</td>
<td>ITYPE</td>
<td>SS1</td>
</tr>
<tr>
<td>32.</td>
<td>LK1</td>
<td>SS2</td>
</tr>
<tr>
<td>33.</td>
<td>LK2</td>
<td>STATS</td>
</tr>
<tr>
<td>34.</td>
<td>LOG</td>
<td>SUBNTK</td>
</tr>
<tr>
<td>35.</td>
<td>LOOKUP</td>
<td>TATOTB</td>
</tr>
<tr>
<td>36.</td>
<td>MBDEF</td>
<td>TECO (MAIN.)</td>
</tr>
<tr>
<td>37.</td>
<td>MBPIK</td>
<td>TEST</td>
</tr>
<tr>
<td>38.</td>
<td>MBROT</td>
<td>TIME</td>
</tr>
<tr>
<td>39.</td>
<td>MBSTFT</td>
<td>TTCALL</td>
</tr>
<tr>
<td>40.</td>
<td>MG1</td>
<td>TPG2</td>
</tr>
<tr>
<td>41.</td>
<td>MG2</td>
<td>TPG4</td>
</tr>
<tr>
<td>42.</td>
<td>MMAV</td>
<td>UFDAT</td>
</tr>
<tr>
<td>43.</td>
<td>MSRSKED</td>
<td>UFDTA</td>
</tr>
<tr>
<td>44.</td>
<td>MSTIME</td>
<td>UFDEST</td>
</tr>
<tr>
<td>45.</td>
<td>MICK</td>
<td>UPORIG</td>
</tr>
<tr>
<td>46.</td>
<td>NETWRK</td>
<td>UPTIME</td>
</tr>
<tr>
<td>47.</td>
<td>NEXT</td>
<td>URES</td>
</tr>
<tr>
<td>48.</td>
<td>OD</td>
<td>VHD4</td>
</tr>
<tr>
<td>49.</td>
<td>OFILE</td>
<td>XBUILD (MAIN.)</td>
</tr>
<tr>
<td>50.</td>
<td>OPEN (main)</td>
<td>XINIT (MAIN.)</td>
</tr>
<tr>
<td>51.</td>
<td>OPEN (subr)</td>
<td>XRUN (MAIN.)</td>
</tr>
<tr>
<td>52.</td>
<td>OZ1</td>
<td>XSETUP (MAIN.)</td>
</tr>
<tr>
<td>53.</td>
<td>OZ2</td>
<td>XSIMU (MAIN.)</td>
</tr>
<tr>
<td>54.</td>
<td>OZFULL</td>
<td>XSTATS (MAIN.)</td>
</tr>
<tr>
<td>55.</td>
<td>PK02</td>
<td></td>
</tr>
</tbody>
</table>
D. Routine Descriptions

BELL produces a single "ding" on the teletype. This routine is used to acquire the DDTS user's attention.

BUFF outputs (to the line printer) the contents of the inter-element vehicle-word transfer buffers. The buffer words are unpacked for ease in interpretation. (This routine is not called by any of the routines in the current version of DDTS; however, it is still contained in the DDTS Library.)

BUILD is one of twelve routines which comprise the BUILD Module — the network construction module (compiler). BUILD is the executive routine of this module, and it offers the user the following options:

1. SD — Set up data to generate files SETUP.DAT, NETWK.DAT, and SGTIE.DAT. These files specify the element configuration (network) and are required by the SETUP Module.

2. OD — This option aids the user in generating files ODMTX.DAT and FNMTX.DAT — these contain the origin/destination probability matrix and the time-varying function identifier matrix, respectively. The latter is an O/D matrix that points to functions used to vary entries in the former.

3. FN — This option allows the user to specify the time-varying functions.
(4) SG -- This option branches to a routine which specifies elements required when the network is specified by an interconnection of subnetworks (interchanges, etc.).

(5) EX (or @) -- This option returns control to the SIMU Module.

BYTE provides a byte addressable array capability. The Fortran call is

\[
\text{CALL BYTE (ARRY, SIZE, BYWD, BYSL, MODE, BUFR)}
\]

where:

- ARRAY is the array name,
- SIZE is the array size (words),
- BYWD is the number bytes/word (1—36),
- BYSL is the byte selected (1 - SIZE*BYWD),
- MODE is '>' or '<' for "to" or "from" buffer, and
- BUFR is a right justified byte.

This routine is used by PPUT to generate a plot image matrix.

CHAIN is the routine used to read in other Modules — while still preserving a permanent resident area (which contains a COMMON area). CHAIN is a DEC (Digital Equipment Corporation) provided program and is described in the DEC-System-10 Mathematical Languages Handbook.

CHECK is a routine of the BUILD Module which checks for improper routing identifiers (associated with an interchange). It types a message when the user specifies more than one direction from an interchange.
as the route to a given destination.

**COMMON** is the Fortran BLOCK DATA program which defines the common storage area. The size of the common area can be changed to accommodate different size networks and PDP-10 core limitations. In the current DDTS System, 34K of core are required. The dimensions of arrays ME, L, and MB limit the maximum number of element groups (currently 160), and the M-array is a general storage area (currently 20,000 words) which limits element parameter values (LK length, SG capacity, etc.) for a given network configuration.

**COKDUO** is a routine for obtaining and releasing core via the PDP-10 Monitor. This routine is used to expand core prior to chaining in a new module and for contracting core after a new module has been chained into core.

**CTEC** is the main program which constitutes the CTEC Module. This program is a modification of TECO, wherein the conversational teletype input to the latter has been replaced by a command file. When CTEC is called by the SIMU-level command CTEC the file CMD.TEC is read in and the TECO commands are executed. If this file is not found on disk, an error message is published. Return to the SIMU Module is accomplished when any TECO exit command (e.g., \$G, or EX$) is encountered in the command file. Illegal commands will result in immediate return to SIMU without further command processing. (A TECO error message will also be published.)
DATA is a RUN-Module routine which outputs time-series data to the disk file (DATA.DAT) every 100 slot-times. The data outputted are OZ vehicle flow counts, number of vehicles in each SG element, and number in each OZ element. This file is used later by routine PLOTS of Module STATS to publish time-series plots.

DATE is a PDP-10 Fortran Library routine which returns today's date (supplied by the Monitor's real-time clock). (The date is placed on the output file in Module STATS and is also used to generate a unique file name when the SIMU-level command RENM is used.)

DELIM is used to separate a string of command words (e.g. SIMU-level commands) which are delimited by commas. Each command word is placed into a separate memory word and hence words are limited to 5 ASCII characters each. Words are left justified and padded with spaces.

DEVFIL is a DDTS utility program (used by several routines) for obtaining device and file specifications from the user. First the message "DEV:FILE =" is published on the TTY (teletype), and then the routine inputs the response of the user. Output from the routine is the device name, logical device number, and the file name. A check is made for illegal device names. A carriage return defaults in the selection of the TTY.

DGl implements Phase 1 for each DG element (diverge), moving a vehicle-word into each element-out-buffer. A DG element segregates vehicle-words according to their destination. A "routing" table,
associated with each DG element, determines which output buffer (node) is selected. As with the MG element the travel time through the DG element is one slot-time.

DG2 is a dummy routine for performing Phase 2 for each DG element. The Phase 2 for each DG element is actually accomplished in DGl where vehicle-words are absorbed from each element-in-buffer.

DUMP is the routine called to execute a SIMU-level DUMP command. It can also be called in the middle of a simulation run (i.e., during the operation of the RUN Module) via the RUN-time command "D". DUMP outputs (to file DUMP.DAT) the following:

1. The date, time, and contents of TITLE.DAT;
2. The SIMU-level command list, and a pointer to command being processed;
3. Current simulation time (ITIME);
4. The number of words used in the M-array (IAMAX);
5. The number of transfers in the NETWK table;
6. The contents of the element-group registers $L(1,ID)$ to $L(8,ID)$ for all active groups;
7. The network-element connections;
8. The current contents of each vehicle-word transfer buffer (in and out buffers); and

The contents of DUMP.DAT are normally observed by PIPing (peripheral-Interchange-Program) them to the Line printer using PIP's "/P" switch. (See DEC's documentation of PIP).
DWORD is a BUILD-Module routine used to generate diverge element routing tables from interchange destination information.

ENTER is a utility routine which produces an entry in the user's disk directory. The file name, extension, and protection are inputs to this routine. An error flag is set if an error condition occurs (as detected by the PDP-10 "ENTER" monitor call).

ENTR is a routine that allows the user to specify a network configuration in an element-by-element manner (option SD of BUILD Module). Care should be exercised in using this option as it is not compatible with all other modules. In general, most network configurations should be specified via SUBNTK (option SB of BUILD Module).

ESTM is the main program of the ESTM Module which is called via the SIMU-level command ESTM or via the monitor-level call RUN DSK: ESTM . Exit from this program brings in the SIMU Module. The ESTM Module is used to estimate the number of groups and the M-array requirements of a given network configuration and set of parameters. The user should call ESTM prior to BUILD to insure that the network model to be constructed is within the core limitations of the particular PDP-10 installation.

EXIT is a PDP-10 Fortran Library routine which is used to terminate execution of DDTS and to return the teletype to monitor command level. (During execution of DDTS a return to the monitor level can also be made by the tC on the teletype.)
EXPND is routine of the SIMU Module used to process SIMU-level command words. Each word is checked for validity, and both abbreviated and macrocommands are expanded. The entire command list is checked by EXPND prior to the dispatch of any commands.

FN is a routine of the BUILD Module which is used to enter functions for time-varying O/D probabilities. The routine first requests the number of functions to be entered, and then for each function it asks for the number of points and their coordinates. After all functions have been entered, the user is given the chance to change any function which may need correction. If no corrections are necessary a carriage return causes file FUNCT.DAT to be outputted and control returns to BUILD.

GATEON is a routine called every simulation slot-time cycle to turn on all SG element gates. The four gates of an SG element group are packed into register L(6,ID). One and zero correspond to "on" and "off", respectively. Routines OZFULL and SGFULL are called immediately after GATEON to turn off the appropriate SG gates (according to the required feedback structure).

GET is used to restore a COMMON area image that has been saved in the disk file SAVE.DAT via routine SAVE. Routine GET is accessed via the SIMU-level command GET. The SIMU-level SAVE and GET commands are used to temporarily save the "state" of DDTS common when a non-chaining routine such as TECO, CTEC, BUILD, ESTM, or PIP is used.
**GRF** is a routine of the ESTM Module that calculates the number of element groups from the number of elements of a given type. For example, 10 LK elements require 3 groups (4 in the first two and 2 in the last).

**IFILE** is a Fortran Library routine which is called to open a DECTape or disk file for reading. The logical unit number and filename are calling parameters. The filename extension assumed is DAT. If the file is not found, an error message is published and a return is made to the PDP-10 Monitor level. The DDTS utility routine LOOKUP can be used prior to a call to IFILE to check for the existence of the file being accessed.

**INIT** is the routine which implements the INIT Module, and is called by the main program XINIT. This routine is used to initialize the state of the simulation to an "empty network" condition. It sets the simulation time to zero and then initializes both M-array and element-transfer-buffer locations with empty-slot vehicle-words. (The origin and destination bytes are set to the ASCII representation of a period.) Some words of the M-array are initialized by the SETUP Module.

**IOACCP** is a utility routine which can be used to input characters via the teletype. If a break character (such as a carriage return) has not been typed, IOACCP waits. Repeated uses of IOACCP returns each of the successive characters of a line. Each character is returned in the low-order seven bits of the single calling para-
IOINTR is a utility routine used to input RUN-Module (RUN-time) commands. The simulation is only interrupted when there is at least one character in the teletype buffer. When routine IOINTR is called, it never puts the job into a wait if a character is not present. When a character is found a flag parameter is set to one, otherwise a zero is returned. The character (if found) is returned in the low-order seven bits of the second calling parameter.

IOTYPE is a utility routine used to output a string of characters. The end of the string is designated by a null word (binary zero).

LK1 performs Phase 1 operations for all LK elements — moving a vehicle-word into each element-out-buffer. This routine uses a segment of the M-array for vehicle-word storage. The movement of vehicles through an LK element is represented by the cyclic positioning of a pointer between lower- and upper-bound positions in the M-array. Vehicle-words moved between the buffers and the M-array do so at the pointer position to the latter. As there is only a single pointer for all LK elements in a group, each must be of the same length.

LK2 performs Phase 2 operations for all LK elements — absorbing a vehicle-word from each element-in-buffer and placing it into an area of the M-array. When the M-array pointer for a given element returns (a fixed simulation time later) to the position at which the vehicle-word was absorbed, Phase 1 (LK1) moves the word to an
element-out-buffer.

LOG provides time-of-day logging information on the teletype so that corresponding time information on titles of files outputted to disk (for later printing on the line printer) may be correlated with the teletype conversation with DDTs. LOG is called prior to executing each SIMU-level command. Only if at least 5 minutes have elapsed since the last log-output to the teletype, is logging information typed. This consists of the time-of-day in 24-hour time, e.g. /13:42/.

LOOKUP checks for the existence of a file in the user's disk area file directory. Input parameters are the filename and extension. If the file is not found a flag parameter is set.

MBDEP is a utility routine used for byte manipulation. The low-order LGH bits of word MRJUST are deposited into word IW at bit position IPNT (the righthand bit) by the Fortran call:

CALL MBDEP (IW, IPNT, LGH, MRJUST).

MBPIK is a utility routine for picking up a byte. With the parameters defined as for routine MBDEP, the Fortran call is

MRJUST = MBPIK (IW, IPNT, LGH).

MBROT is a utility routine for doing word rotations. The Fortran call is

IWROT = MBROT (IW, NDIGPL),

where IWROT is the word IW rotated NDIGPL places to the left. (If NDIGPL is negative a right-rotation occurs.) All 36 bits of the
word are rotated.

MBSFT is similar to MBROT except it does a word shift rather than a rotate. The call is

\[ IWSFT = MBSFT(IW, NDIGPL) \]

On a left shift, for example, bits are lost from the left end of the word and zeroes are added to the right end.

MGl performs Phase 1 for all MG elements — moving a vehicle-word into each element-out-buffer. A merge element is used to combine two vehicle-words, absorbed from two input buffers. If both input words represent a vehicle then a "collision" error flag is set (4 ERR), the error trap in the RUN Module will publish a message, and one of the vehicle-words is destroyed. Travel time through a merge element is always one slot time.

MG2 is a dummy routine called at Phase 2 time. Phase 2 for MG elements is actually implemented in MGl.

MMADVN is a routine to advance the pointer in a merge memory array (every slot-time). Each bit in the merge memory for a particular merge represents the presence or absence of a vehicle. The array must be large enough to accommodate an entry for all vehicle-words which are scheduled to pass through the merge. (These are held in the OZ elements which "feed" the interchange containing the merge.)

MMSKED is a routine called by OZ2 to schedule a vehicle through a merge point. When a call is made to MMSKED, a scan is made
through the merge memory array associated with the approached
merge. The first empty slot is found (0-bit) and the bit is
set to 1 — representing a reservation. (See MMADVN routine.)

MSTIME simply returns the time-of-day in milliseconds after midnight.

MUCK can be called to do a setup so that DDTS can handle error traps
normally handled by the Fortran Operating System. It also
shortens the exiting routine message. MUCK is not called in the
current version of DDTS.

NETWK is the routine called between Phase 1 and Phase 2 to transfer
vehicle-words from element-out-buffers to element-in-buffers.
The Network Transfer Table (file NETWK.DAT) specifies the element
connections. Improved efficiency is achieved by moving the
instructions, which actually accomplish the word-by-word transfers,
to fast memory (the PDP-10 accumulators) prior to looping. This
reduces the instruction "fetch" times.

NEXT implements the SIMU-level command "NEXT." This routine makes the
next set of input files to the SETUP Module available. The files
selected for processing are stacked by the "QUEUE" command (which
creates file QUEUE.DAT). Routine NEXT reads in file QUEUE.DAT,
picks up the filenames off the top of each stack (there is a
stack for each of the files required by the SETUP Module), and
renames these files to the names required by Module SETUP. The
routine also updates the pointers to each stack in file QUEUE.DAT.
See routine QUEUE.

OD implements the OD option of the BUILD Module. It is used to generate matrix files ODMTX and FNMTX. Matrices may be specified element-by-element or special suboptions are available for creating balanced demand matrices (all entries are the same but zeroes are on the diagonal). The user is given a chance to change single entries before the matrix is written onto disk. The TECO Module may also be used to create matrix files and to alter same.

OFILE is called to open (or create) a file directory entry to enable outputting to the user's disk area. The input parameters are the logical unit number and the filename. The extension DAT is assumed.

OPEN (Main program) is a routine which implements the OPEN Module. When the SIMU-level command "OPEN" is used, the SIMU Module chains in the OPEN Module (leaving the COMMON core intact), which contains the DDT routine, and control is then transferred to the latter.

DDT (Dynamic Debugging Technique) is described in the DEC System-10 Assembly Language Handbook. The user may enter DDT commands for the purpose of examining and modifying COMMON area storage words. This is useful in debugging changes to DDTS and in simulating anomalous conditions (by altering vehicle-words, etc.). To return to the SIMU Module the user must either enter the DDT command:

\[ \text{JRST 2,} \text{JOPOPOC}\$X \]

or type \text{C} to return to the Monitor, then type REEN followed by a
carriage return (which results in OPEN chaining in SIMO).

OPEN (Subroutine) is a utility routine of the BUILD Module which locates the position of the highest-order non-zero bit in a storage word.

OZ1 is the Phase 1 routine for all OZ elements. It transfers vehicle-words to the element-out-buffer area when the scheduled OZ departure time has come up (for each vehicle-word in the OZ element storage area).

OZ2 is the Phase 2 routine for all OZ elements. It absorbs vehicle-words from the element-in-buffer area and schedules an element departure time prior to placing the vehicle-word in an OZ element storage area. If an OZ element cannot accommodate a vehicle-word it is lost and a "3 ERR" occurs.

OZFULL is the routine that implements OZ-to-SG feedback when each OZ element has more than a given number of vehicles. (This number is set by the threshold associated with each OZ element.) This routine assumes that the ordering of the SG elements matches that for the OZ elements, and that each SG element feeds the corresponding OZ element. A call to OZFULL is made once each simulation time, and after calling GATEON.

PK02 is a utility routine used to pack the low-order 16 bits of two consecutive words into a single word. It publishes an error message if a byte is over 16 bits long.
PKØ4 is similar to PKØ2 except that four 8-bit bytes are packed.
It publishes an error message if a byte is over 8 bits long.

PKDATA packs data into the data-byte position of a vehicle word (bits 14-20).

PKDEST packs an ASCII character into the destination-byte position of a vehicle-word (bits 0-6).

PKORIG packs an ASCII character into the origin-byte position of a vehicle word (bits 7-13).

PKTIME is used to pack the time a vehicle enters the system (is generated) into the time-byte position of a vehicle-word (bits 21-35).

PKUP is a utility routine for equal-length byte packing and unpacking.

The Fortran call is

CALL PKUP(MODE, IWORD, IARRAY, NBYTE)

where

- MODE is 'PK' for packing and 'UP' for unpacking;
- IWORD is the packed word or word to be packed;
- IARRAY is an array with the byte information right justified; and

NBYTE is the number of bytes to be processed.

The number of bits per byte is a function of the number of bytes per word. The largest possible byte size is used (which results in equal length bytes). The packed bytes are left justified if there are any unused bits. For example if NBYTE is equal to 5, then there will be 7 bits per byte and the rightmost bit of a
PLOTS is a routine of the STATS Module which reads in data from the file DATA.DAT (which is prepared by the RUN Module) and adds line-printer plots to disk file GOODS.DAT. The data plotted are flow densities (measured at the output node of each OZ element). Data for all elements in a given group are plotted on the same page. PLOTS calls routines PPUT and POUT. The M-array area is used to store the plotted images.

POUT adds borders and scaling information to a line-printer plot and then outputs the core image of the plot to a disk file.

PPUT adds points to the core image of a line-printer plot. The core image array, the coordinates, and ASCII character (to be plotted) are input parameters to this routine. A single point is plotted per call.

QUEUE implements the SIMU-level "QUEUE" command. QUEUE generates the QUEUE.DAT disk file which is used by the NEXT routine to select the next set of files for processing by the SETUP Module. A string of file names (delimited by commas) are inputted to this routine when it asks for each of four types of files (ODMTX, FNMTX, FUNCT, and NETWK). The names given for NETWK also apply to the two other network-specifying-files, SETUP and SGTIE. A lookup is made to see that each file specified is in the user's disk area. The extensions ODX, FNX, FNC, NET, SET, and FBK are assumed for the
respective files. Routine NEXT selects files of each type by going through the inputted file strings and renaming the appropriate disk file to the name accepted by the SETUP Module. When the end of the string is reached, the last entry is reused. If there are no filenames in a string the filenames accepted by the SETUP Module are assumed.

**RAN** is Fortran Library routine for generating pseudo-random numbers.

RAN returns a floating-point number in the range 0<X<1. (This number is the conversion of a generated random integer.) The general form of the pseudo random number generator is

\[ X_{n+1} = K X_n \mod m \]

where

\[ m = 2^{31} - 1 \]
\[ K = 14^{29} \]

If the generated random integer is larger than 27 bits, it is truncated to 27 bits prior to floating-point conversion and normalization. **RAN** is called by routine SS1 to generate vehicle trip patterns according to a time-varying demand matrix. The SIMUL-level command "RSEED" can be used to reset the random number seed (initial "random" number).

**REEN** is a routine called in the initialization stage of most of the modules of DDTS for the purpose of setting up reentry addresses for the resident module. If a user wishes to abort the execution of a particular module via a \(^{1}C\) on the teletype, he may reenter the resident module of DDTS by using the monitor-level command "REEN".
This command results in control being transferred to the "REEN
handler" section of the REEN routine. The line "WHERE:" is typed
and the user must respond with a single alphabetic character —
representing the desired reentry point (no carriage return is
necessary). The reentry points for each module are defined in the
DDTS User's Manual. An illegal reentry selection results in the
message "...NO GOOD..." and the routine waits for a valid entry.

REGS is the routine called by DUMP to output element–group register
information. It outputs in octal format the contents of each
register, L(1,ID) through L(8,ID), for each element group (ID).
Element groups of a given type are outputted together and the
output order is LK, SG, OZ, SS, MG, DG.

RELEAS is a Fortran Library routine for closing out I/O on a device
initialized by IFILE or OFILE. In DDTS this operation is normally
accomplished by calling IFILE or OFILE (which do a RELEAS if
necessary prior to opening the selected channel).

RENAME is a utility routine for both renaming disk files and/or changing
their protection. Calling parameters are the old name, old exten-
sion, new name, new extension and protection code (a zero results
in the standard system protection). There is an error return when
the old file is not found or the new filename is already in use.
If the new filename and extension are words containing all "space"
characters the file will be deleted.
RENM is a routine called by the SIMU Module to execute the "RENM" command. RENM causes the file outputted by the STATS Module (GOODS.DAT) to be renamed to a unique filename. The filename is of the form:

```
  ddmnn.RUN
```

where

- `dd` is a two digit day designator.
- `mm` is a two letter month designator, and
- `n` is a unique run number.

(RENM scans through the user's disk directory to insure that a name is not reused.) For example, the filename for the third run on Sept. 17 might be

```
  17SP2.RUN
```

RENM will also ring the teletype bell three times and output the message:

```
RENM< 800 OUTFILE: 17SP2.RUN >
```

RENM is useful when several runs are made in succession — without user intervention.

RESET is a Fortran Library routine which resets I/O operations on all devices. The Fortran Compiler automatically generates a call to RESET at the beginning of each main (MAIN.) program. RESET clears tables and flags in the Fortran Operating System (FORSE.). The latter handles all Fortran I/O.

RING is a utility routine for audible signaling to the user via the teletype bell. There are four calling parameters to control ringing duration, number of sets of rings, etc.
RNAME is a BUILD-Module routine used for renaming files SETUP.DAT, NETWK.DAT, and SGTIE.DAT to name.SET, name.NET, and name.FBK, where "name" is a filename selected by the user for a network he has just constructed. An error message is outputted if a filename is already in use. If the user does not enter a name (i.e., responds with a carriage return), no files will be renamed.

RUN is the routine which implements the RUN Module. RUN (and routines which it calls) performs the simulation algorithm — generating, processing, and destroying entities called "vehicle-words". RUN calls the following routines once each simulated slot-time:

- UPDAT
- GATEON
- OZFLL
- SGFULL
- LK1
- SG1
- OZ1
- SS1
- MG1
- DG1
- NETWK
- LK2
- SG2
- OZ2
- SS2
- MG2
- DG2
- DATA
- TEST

RUNCMD is a RUN-Module routine called once every 100 slot-times for the purpose of dispatching RUN-time commands which are picked up from the user's teletype input buffer.

RUNTIM is a utility routine which returns the job's run-time in milliseconds. This routine is used by the RUN Module to determine the run-time for a given simulation. This information is automatically outputted to the DDTS user's teletype at the end of each run (i.e., upon exit from Module RUN).

RUNUOO is a utility routine for program execution of the monitor-level command:

```
.RUN DEV:PROG
```
This routine results in the saved program, PROG, on device, DEV, being loaded and executed. This routine is used to read in and branch to non-chaining modules. The Fortran call is

```fortran
CALL RUNUOO(DEV,PROG)
```

If the selected program is not found, an error message will be published and the program will return to the monitor level.

SAVE is a SIMU-Module routine called by the "SAVE" command to save the COMMON core area. (See explanation of Routine GET.)

SAVRAN is a Fortran Library routine used to save the last random integer generated by a call to RAN. SAVRAN is called at the end of a simulation run to save the state of the random number generator in COMMON core prior to chaining in Module SIMU.

SETRAN (a Fortran Library routine) is called at the beginning of a simulation run to set the random number generator seed. This seed is either the last random number generated in the previous run, the standard starting seed, or a seed selected by the user via the SIMU-level "SET" command.

SETUP is a routine used to implement the SETUP Module. The inputs required by SETUP are the files SETUP, NETWK, ODMTX, FNMTX, FUNCT, AND SGTIE (all with extension DAT). First Routine SETUP checks for the existence of each of these files, and if any are not found, a message is published and an error return is made to the SIMU Module. SETUP then reads in each file and uses the information to setup
or assemble the COMMON area of core so that a simulation may be run. Several conditions are checked by SETUP to ensure that the network being simulated is within some of the limitations of DDTS.

**SG1** is the Phase 1 routine for the SG elements. It moves a vehicle-word into each element-out-buffer. This word can represent either a vehicle or an empty slot. The release of a vehicle is controlled by a gate associated with each SG element.

**SG2** is the Phase 2 routine for the SG elements. It absorbs vehicle-words from each element-in-buffer. If an SG element exceeds its capacity the entering vehicle-word will be lost, and a "2 ERR" will be generated unless the SG element was fed by an SS element. In the latter case the appropriate SS element reject count will be incremented by one.

**SGTIE** is a BUILD Module routine which is called by SUBNTK to generate the SG-toSG element feedback table. For each SG element this routine finds all the SG elements on the immediate upstream interchange (if such exists) and then generates the appropriate table entry.

**SGTIE2** further processes the table generated by SGTIE. It converts the Element-Register references in the table from double subscript to equivalent single subscript addresses. (This is to increase RUN-time efficiency.) It packs this information along with required byte addressing information and then outputs file SGTIE.DAT.

**SSI** is the Phase 1 routine for the SS elements. It is the source portion
of the element, and thus generates a vehicle-word (which may represent an empty slot) every simulated slot-time. SSI uses the pseudo random number generator RAN and an O/D probability table to generate vehicle destinations. SSI places the generated vehicle-words in the appropriate element-out-buffers.

SS2 is the Phase 2 routine for the SS elements. It performs the "sink" portion of the SS element function. Each slot-time vehicle-words are absorbed and data from same are unpacked and utilized to update accumulated statistic tables. These tables reside in COMMON core, and are later used by the STATS Module to calculate various O/D statistic matrices (mean delay, standard deviation of delay, mean flow, etc.).

STATS is the routine which implements the STATS Module. (It is called by the main program "driver" routine XSTATS.) The input to STATS is the data placed into COMMON core by the RUN Module and the time-series flow data found in file DATA.DAT. STATS also refers to file TITLE.DAT (if the file exists) for titling purposes. The output from Module STATS is the file GOODS.DAT which contains the results of the last simulation. This file can be renamed to a unique name via the SIMU-level command "RENM". The file is outputted to the line printer using DEC's PIP Program with the switch option "/P".
SUBNTK implements the "SB" option of the BUILD Module. SUBNTK saves the user time in setting up large networks by expanding user specifications of subnetwork interconnection into the appropriate collection of simple elements. The types of subnetworks available are:

- 4W — A four-way interchange;
- 3W — A three-way interchange;
- EE — An Entrance-Exit facility (A 3W with an SS pair);
- SC — A two-way guideway section; and
- ON — A SS element.

Besides specifying subnetwork types, the user specifies their interconnection via node numbers and routing procedures via alphabetic letters. When entering subnetwork types the user may control BUILD via the following subcommands:

- RB — Rubout last entry and allow user to reenter it;
- ER — Erase a selected entry and allow the use to replace it;
- LT — List the subnetworks which have been specified; and
- WR — Wrap up, write files, and return to routine BUILD.

See the DDTS User's Manual for a description of the teletype interaction required to construct a network.
TATOTB is a transfer according to table routine previously used for element vehicle-word buffer transfers. It is still in the DDTS Library; however, TATOTB's function is currently handled by routine NETWRK (a routine which incorporates the PDP-10's fast memory for instruction storage).

TECO is a program which comprises the TECO Module of DDTS. This program is a modified version of DEC's TECO (Text Editor and Corrector) Program. The following addition commands have been added to the latter (which is documented in the DEC-10 User's Handbook):

- `tRname(SS)`: Run the program called "name" (it will be found in the user's disk area);
- `tSname(SS)`: Run the system program called "name", e.g., `tSPlHD®` (Note: ∼S is a Altmode character); and
- `@`: "@" at the beginning of a line and followed by any characters will cause a return to the SIMU Module when the first "break" character such as (carriage return) or "S" (altmode) is encountered. "@" is interpreted as a normal TECO command when found in any other position.

TECO is a utility module of DDTS as it can be used to create, change, or correct many of the files used in DDTS. For example, TECO can be used to create the "title file" TITLE.DAT, to modify the "SG-feedback file" SGTIE.DAT, or to alter the format of the STATS Module output file, GOODS.DAT.

TEST is the RUN Module simulation error trapping program which publishes the error conditions detected in the Phase 1 and Phase 2 routines.
It publishes (on the TTY) the error type and the simulation slot-time at which the error was detected. (An error location is also given but this is only of historic importance.) If more than one type of error is detected during a single slot-time interval, only the last error type found is published. Routine TEST also checks to see that all entries in the network vehicle-word transfer table are valid. If they are not an error message is published (BAD TABLE AT ...).

TIME is a PDP-10 Fortran Library routine used to obtain the current time (in hours, minutes, seconds, and tenths past midnight).

TTCALL is a Fortran accessible utility routine for using the PDP-10 "monitor call" termed TTCALL. (See the DEC System-10 Assembly Language Handbook.) This routine enables for special teletype I/O functions not normally possible with Fortran.

UP02 is a utility routine for unpacking two-byte words (packed by PK02).

UP04 is a utility routine for unpacking four-byte words (packed by PK04).

UPDAT is a routine called by the RUN routine to update the time-dependent O/D probabilities. Every 100 slot-times it takes the (core stored) ODMTX and multiplies the entries by the appropriate functions. (These functions are ones selected from FUNCT using the FNMTX.) This matrix is then "row integrated" so that it is in
the form required by the vehicle-word generator in SSL.

UPDATA unpacks data from the data-byte position of a vehicle-word (bits 14-20).

UPDEST unpacks an ASCII character from the destination-byte position of a vehicle-word (bits 0-6).

UPORIG unpacks an ASCII character from the origin-byte position of a vehicle-word (bits 7-14).

UPTIME is used to unpack the time a vehicle entered the system from the time-byte position (bits 21-35) of its vehicle-word.

VHWD is a general vehicle-word pack/unpack routine. It performs the same functions (although not as efficiently) as the specialized vehicle-word packing and unpacking routines (PKDATA, UPDATA, etc).

XBUILD is the main program for Module BUILD. It sets up REEN points and then calls subroutine BUILD. The name used to label a Fortran main program such as XBUILD is "MAIN."

XINIT is the main program for Module INIT. It sets up REEN points and then calls subroutine INIT.

XRUN is the main program for Module RUN. It calls subroutine RUN.

XSETUP is the main program for Module SETUP. It calls subroutine SETUP.
XSIMU is the main program for the SIMU Module. This module is the executive module of the DDTS system. All other modules are loaded and called from the SIMU Module. When any other module completes its function it returns to SIMU so that the next SIMU-level command can be dispatched.

XSTATS is the main program for the Module STATS. XSTATS sets up REEN points and then calls subroutine STATS.


43. Wilson, David Gordon (editor), Automated Guideway Transportation Between and Within Cities, to be published by the Urban Systems Laboratory, Massachusetts Institute of Technology.

44. Wolf, Robert A., Metrotran - 2000, Transportation Research Department, Report CAL No. 150, Cornell Aeronautical Laboratory, Inc., Cornell University, Buffalo, New York, October 1967.