ENHANCED TRACKING OF AIRBORNE TARGETS USING FORWARD LOOKING INF--ETC(U)

S K ROGERS

AFIT/GE/EE/81D-5
ENHANCED TRACKING OF AIRBORNE TARGETS USING FORWARD LOOKING INFRARED MEASUREMENTS

THESIS

AFIT/GEO/EE/81D-5

Steven K. Rogers

1st Lt

USAF
ENHANCED TRACKING OF AIRBORNE 
TARGETS USING FORWARD 
LOOKING INFRARED MEASUREMENTS 

THESIS 

Presented to the Faculty of the School of Engineering 
of the Air Force Institute of Technology 
Air University 
in Partial Fulfillment of the 
Requirements for the Degree of 
Master of Science in Electrical Engineering 

by 
Steven K. Rogers B.S.E.E. 
1st Lt USAF 
Graduate Electro-Optics 
December 1981
Preface

This study was part of a continuing effort to design next generation trackers for one of the Air Force Weapons Laboratory's laser weapons using modern optimal estimation and control techniques. This report uses these techniques to develop an extended Kalman filter which uses outputs of a forward looking infrared sensor in a tracking problem. As an alternate tracker, an extended Kalman filter is designed to process the position estimate outputs of a correlation algorithm.

I wish to express my thanks to Dr. Peter S. Maybeck, my advisor, for his motivation, expert advice and valuable time. Mr. David R. McGrew deserves additional thanks. His knowledge of fundamental engineering concepts proved invaluable to the development and testing of the data processing algorithms used in this research. I wish also to thank James Emett "Microman" Crider and Robert Suizu for their assistance in using the Modcomp minicomputer.

The author reserves a very special thanks to his father who continually gives him an unlimited source of inspiration. This thanks is also extended to his mother as the proverbial source of tender loving care. To his wife, whose support was absolutely essential for the completion of all his studies from the early days of midnight factory shifts and daytime classes to today, the author has only his never ending and
passionate love. To my beautiful baby daughter, Sarah, it certainly is nice to always have someone to come home to who will give me a pretty dimpled smile. Last but definitely most, to his primary source of pride and everlasting joy, his son, Jeremiah, who always had to wait for his Dad to come home or finish studying before they could go out to play together, the author has his infinite love and sincerest apologies. I would also like to acknowledge my typist, Ms Cheryl Nicol. Her professionalism and dedication made completion of this document in time to meet an early deadline possible and was greatly appreciated.

Steven K. Rogers
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>xiii</td>
</tr>
<tr>
<td>Abstract</td>
<td>xviii</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>2</td>
</tr>
<tr>
<td>Problem</td>
<td>4</td>
</tr>
<tr>
<td>Plan of Attack</td>
<td>9</td>
</tr>
<tr>
<td>Overview</td>
<td>12</td>
</tr>
<tr>
<td>II. Derivation of the Nonlinear and Linearized Intensity Functions</td>
<td>14</td>
</tr>
<tr>
<td>Introduction</td>
<td>14</td>
</tr>
<tr>
<td>Pattern Recognition</td>
<td>14</td>
</tr>
<tr>
<td>Two-Dimensional Fourier Transforms</td>
<td>16</td>
</tr>
<tr>
<td>Shifting Property of Fourier Transforms</td>
<td>18</td>
</tr>
<tr>
<td>Smoothing in the Fourier Domain</td>
<td>20</td>
</tr>
<tr>
<td>Derivative Property of Fourier Transforms</td>
<td>21</td>
</tr>
<tr>
<td>Derivation of Intensity Functions</td>
<td>23</td>
</tr>
<tr>
<td>III. Truth Model</td>
<td>25</td>
</tr>
<tr>
<td>Introduction</td>
<td>25</td>
</tr>
<tr>
<td>Target Dynamics Model</td>
<td>25</td>
</tr>
<tr>
<td>Atmospheric Jitter Model</td>
<td>27</td>
</tr>
<tr>
<td>State Space Model</td>
<td>29</td>
</tr>
<tr>
<td>Propagation Equations</td>
<td>32</td>
</tr>
<tr>
<td>Measurements</td>
<td>34</td>
</tr>
<tr>
<td>Spatially Correlated Background Noise</td>
<td>36</td>
</tr>
<tr>
<td>Summary of Truth Model</td>
<td>40</td>
</tr>
<tr>
<td>IV. Extended Kalman Filter</td>
<td>41</td>
</tr>
<tr>
<td>Introduction</td>
<td>41</td>
</tr>
<tr>
<td>Target Dynamics and Atmospheric Jitter Filter Models</td>
<td>41</td>
</tr>
<tr>
<td>State Space Model</td>
<td>42</td>
</tr>
</tbody>
</table>
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Propagation</td>
<td>44</td>
</tr>
<tr>
<td>Measurement Update Equations</td>
<td>47</td>
</tr>
<tr>
<td>Summary of the Extended Kalman Filter Equations</td>
<td>50</td>
</tr>
<tr>
<td>V. Correlator-Kalman Filter Tracker</td>
<td>52</td>
</tr>
<tr>
<td>Introduction</td>
<td>52</td>
</tr>
<tr>
<td>Correlator Implementation</td>
<td>53</td>
</tr>
<tr>
<td>Correlator Error Statistics</td>
<td>63</td>
</tr>
<tr>
<td>Kalman Filter Tracker</td>
<td>68</td>
</tr>
<tr>
<td>VI. Performance Analysis</td>
<td>71</td>
</tr>
<tr>
<td>Introduction</td>
<td>71</td>
</tr>
<tr>
<td>Derivation of Intensity Functions</td>
<td>72</td>
</tr>
<tr>
<td>Tracking Ability</td>
<td>75</td>
</tr>
<tr>
<td>Variation of Parameters</td>
<td>77</td>
</tr>
<tr>
<td>Plotting Results</td>
<td>81</td>
</tr>
<tr>
<td>Table Summary</td>
<td>99</td>
</tr>
<tr>
<td>Summary</td>
<td>112</td>
</tr>
<tr>
<td>VII. Optical Processing Alternatives</td>
<td>113</td>
</tr>
<tr>
<td>Introduction</td>
<td>113</td>
</tr>
<tr>
<td>Background</td>
<td>114</td>
</tr>
<tr>
<td>Applying Optical Processing to Tracking</td>
<td>118</td>
</tr>
<tr>
<td>Conclusion</td>
<td>123</td>
</tr>
<tr>
<td>VIII. Conclusions and Recommendations</td>
<td>125</td>
</tr>
<tr>
<td>Conclusions</td>
<td>125</td>
</tr>
<tr>
<td>Recommendations</td>
<td>126</td>
</tr>
<tr>
<td>Bibliography</td>
<td>129</td>
</tr>
<tr>
<td>Appendix A: Two-Dimensional Finite Discrete Fourier Transform Interpretation and Test</td>
<td>131</td>
</tr>
<tr>
<td>Appendix B: Shifting Property of the Two-Dimensional Finite Discrete Fourier Transform</td>
<td>146</td>
</tr>
<tr>
<td>Appendix C: Implementation of the Exponential Smoothing Algorithm</td>
<td>151</td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix D:</td>
<td>Implementation of the Spatial Derivatives Using Fourier Transforms</td>
<td>155</td>
</tr>
<tr>
<td>Appendix E:</td>
<td>Generation of White Gaussian Noise Process</td>
<td>161</td>
</tr>
<tr>
<td>Appendix F:</td>
<td>Discrete Representation of the Derivative of the Intensity Profile with Respect to the Kalman Filter States</td>
<td>164</td>
</tr>
<tr>
<td>Appendix G:</td>
<td>Monte Carlo Study</td>
<td>169</td>
</tr>
<tr>
<td>Appendix H:</td>
<td>Computer Software</td>
<td>171</td>
</tr>
<tr>
<td>Appendix I:</td>
<td>Plots of Tracking Errors</td>
<td>269</td>
</tr>
<tr>
<td>Appendix J:</td>
<td>Kalman Filters</td>
<td>334</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>Data Processing Algorithm</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Research Plan of Attack Flow Diagram</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Third Order Shaping Filter</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>FLIR Data Frame Pixel Numbering Scheme</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>First and Second Nearest Neighbor</td>
<td>38</td>
</tr>
<tr>
<td>6</td>
<td>Centered Single Gaussian Template</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>Noise Corrupted Data Array</td>
<td>57</td>
</tr>
<tr>
<td>8</td>
<td>Result of Cross Correlation of Data and Template</td>
<td>58</td>
</tr>
<tr>
<td>9</td>
<td>Diagonal Quadrant Swap of Cross Correlation</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>Result of Thresholding of .5</td>
<td>62</td>
</tr>
<tr>
<td>11a</td>
<td>Histogram of Error in Horizontal Position Estimate</td>
<td>65</td>
</tr>
<tr>
<td>11b</td>
<td>Histogram of Error in Vertical Position Estimate</td>
<td>66</td>
</tr>
<tr>
<td>12</td>
<td>X Minus Errors</td>
<td>84</td>
</tr>
<tr>
<td>13</td>
<td>Y Minus Errors</td>
<td>85</td>
</tr>
<tr>
<td>14</td>
<td>X Plus Errors</td>
<td>86</td>
</tr>
<tr>
<td>15</td>
<td>Y Plus Errors</td>
<td>87</td>
</tr>
<tr>
<td>16</td>
<td>X Position Error</td>
<td>88</td>
</tr>
<tr>
<td>17</td>
<td>Y Position Error</td>
<td>89</td>
</tr>
<tr>
<td>18</td>
<td>X Centroid Minus Error</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>Y Centroid Minus Error</td>
<td>91</td>
</tr>
<tr>
<td>20</td>
<td>X Centroid Plus Error</td>
<td>92</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Y Centroid Plus Error</td>
<td>93</td>
</tr>
<tr>
<td>22</td>
<td>X Centroid Position Error</td>
<td>94</td>
</tr>
<tr>
<td>23</td>
<td>Y Centroid Position Error</td>
<td>95</td>
</tr>
<tr>
<td>24</td>
<td>Error of Estimated H</td>
<td>96</td>
</tr>
<tr>
<td>25</td>
<td>Error of Estimated DH/DX</td>
<td>97</td>
</tr>
<tr>
<td>26</td>
<td>Error of Estimated DH/DY</td>
<td>98</td>
</tr>
<tr>
<td>27</td>
<td>Basic Optical Processing System</td>
<td>115</td>
</tr>
<tr>
<td>E-1</td>
<td>Output Probability Density of Pseudorandom Code</td>
<td>161</td>
</tr>
<tr>
<td>F-1a</td>
<td>Continuous One-Dimensional Gaussian</td>
<td>166</td>
</tr>
<tr>
<td>F-1b</td>
<td>Discrete One-Dimensional Gaussian</td>
<td>166</td>
</tr>
<tr>
<td>F-2</td>
<td>Shifted Intensity Profile</td>
<td>167</td>
</tr>
<tr>
<td>F-3</td>
<td>Intensity's Spatial Derivative</td>
<td>167</td>
</tr>
<tr>
<td>F-4</td>
<td>Derivative with Respect to the States</td>
<td>168</td>
</tr>
<tr>
<td>G-1</td>
<td>Generation of Error Samples</td>
<td>169</td>
</tr>
<tr>
<td>I-1</td>
<td>Pad Zeros X Minus Errors</td>
<td>271</td>
</tr>
<tr>
<td>I-2</td>
<td>Pad Zeros Y Minus Errors</td>
<td>272</td>
</tr>
<tr>
<td>I-3</td>
<td>Pad Zeros X Plus Errors</td>
<td>273</td>
</tr>
<tr>
<td>I-4</td>
<td>Pad Zeros Y Plus Errors</td>
<td>274</td>
</tr>
<tr>
<td>I-5</td>
<td>Pad Zeros X Position Errors</td>
<td>275</td>
</tr>
<tr>
<td>I-6</td>
<td>Pad Zeros Y Position Errors</td>
<td>276</td>
</tr>
<tr>
<td>I-7</td>
<td>Pad Zeros X Centroid Minus Errors</td>
<td>277</td>
</tr>
<tr>
<td>I-8</td>
<td>Pad Zeros Y Centroid Minus Errors</td>
<td>278</td>
</tr>
<tr>
<td>I-9</td>
<td>Pad Zeros X Centroid Plus Errors</td>
<td>279</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-10</td>
<td>Pad Zeros Y Centroid Plus Errors</td>
<td>280</td>
</tr>
<tr>
<td>I-11</td>
<td>Pad Zeros X Centroid Position Errors</td>
<td>281</td>
</tr>
<tr>
<td>I-12</td>
<td>Pad Zeros Y Centroid Position Errors</td>
<td>282</td>
</tr>
<tr>
<td>I-13</td>
<td>Pad Zeros Error of Estimated $h$</td>
<td>283</td>
</tr>
<tr>
<td>I-14</td>
<td>Pad Zeros Error of Estimated $\frac{DH}{DX}$</td>
<td>284</td>
</tr>
<tr>
<td>I-15</td>
<td>Pad Zeros Error of Estimated $\frac{DH}{DY}$</td>
<td>285</td>
</tr>
<tr>
<td>I-16</td>
<td>Removing Two Highest Spatial Frequencies $X$ Minus Errors</td>
<td>287</td>
</tr>
<tr>
<td>I-17</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Minus Errors</td>
<td>288</td>
</tr>
<tr>
<td>I-18</td>
<td>Removing Two Highest Spatial Frequencies $X$ Plus Errors</td>
<td>289</td>
</tr>
<tr>
<td>I-19</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Plus Errors</td>
<td>290</td>
</tr>
<tr>
<td>I-20</td>
<td>Removing Two Highest Spatial Frequencies $X$ Position Errors</td>
<td>291</td>
</tr>
<tr>
<td>I-21</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Position Errors</td>
<td>292</td>
</tr>
<tr>
<td>I-22</td>
<td>Removing Two Highest Spatial Frequencies $X$ Centroid Minus Errors</td>
<td>293</td>
</tr>
<tr>
<td>I-23</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Centroid Minus Errors</td>
<td>294</td>
</tr>
<tr>
<td>I-24</td>
<td>Removing Two Highest Spatial Frequencies $X$ Centroid Plus Errors</td>
<td>295</td>
</tr>
<tr>
<td>I-25</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Centroid Plus Errors</td>
<td>296</td>
</tr>
<tr>
<td>I-26</td>
<td>Removing Two Highest Spatial Frequencies $X$ Centroid Position Errors</td>
<td>297</td>
</tr>
<tr>
<td>I-27</td>
<td>Removing Two Highest Spatial Frequencies $Y$ Centroid Position Errors</td>
<td>298</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>I-28</td>
<td>Removing Two Highest Spatial Frequencies Error of Estimated ( h )</td>
<td>299</td>
</tr>
<tr>
<td>I-29</td>
<td>Removing Two Highest Spatial Frequencies Error of Estimated ( \frac{Dh}{Dx} )</td>
<td>300</td>
</tr>
<tr>
<td>I-30</td>
<td>Removing Two Highest Spatial Frequencies Error of Estimated ( \frac{Dh}{Dy} )</td>
<td>301</td>
</tr>
<tr>
<td>I-31</td>
<td>Removing Four Highest Spatial Frequencies X Minus Errors</td>
<td>303</td>
</tr>
<tr>
<td>I-32</td>
<td>Removing Four Highest Spatial Frequencies Y Minus Errors</td>
<td>304</td>
</tr>
<tr>
<td>I-33</td>
<td>Removing Four Highest Spatial Frequencies X Plus Errors</td>
<td>305</td>
</tr>
<tr>
<td>I-34</td>
<td>Removing Four Highest Spatial Frequencies Y Plus Errors</td>
<td>306</td>
</tr>
<tr>
<td>I-35</td>
<td>Removing Four Highest Spatial Frequencies X Position Errors</td>
<td>307</td>
</tr>
<tr>
<td>I-36</td>
<td>Removing Four Highest Spatial Frequencies Y Position Errors</td>
<td>308</td>
</tr>
<tr>
<td>I-37</td>
<td>Removing Four Highest Spatial Frequencies X Centroid Minus Errors</td>
<td>309</td>
</tr>
<tr>
<td>I-38</td>
<td>Removing Four Highest Spatial Frequencies Y Centroid Minus Errors</td>
<td>310</td>
</tr>
<tr>
<td>I-39</td>
<td>Removing Four Highest Spatial Frequencies X Centroid Plus Errors</td>
<td>311</td>
</tr>
<tr>
<td>I-40</td>
<td>Removing Four Highest Spatial Frequencies Y Centroid Plus Errors</td>
<td>312</td>
</tr>
<tr>
<td>I-41</td>
<td>Removing Four Highest Spatial Frequencies X Centroid Position Errors</td>
<td>313</td>
</tr>
<tr>
<td>I-42</td>
<td>Removing Four Highest Spatial Frequencies Y Centroid Position Errors</td>
<td>314</td>
</tr>
<tr>
<td>I-43</td>
<td>Removing Four Highest Spatial Frequencies Error of Estimated ( h )</td>
<td>315</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>I-44</td>
<td>Removing Four Highest Spatial Frequencies Error of Estimated Dh/Dx</td>
<td>316</td>
</tr>
<tr>
<td>I-45</td>
<td>Removing Four Highest Spatial Frequencies Error of Estimated Dh/Dy</td>
<td>317</td>
</tr>
<tr>
<td>I-46</td>
<td>Maximum Noise X Minus Errors</td>
<td>319</td>
</tr>
<tr>
<td>I-47</td>
<td>Maximum Noise Y Minus Errors</td>
<td>320</td>
</tr>
<tr>
<td>I-48</td>
<td>Maximum Noise X Plus Errors</td>
<td>321</td>
</tr>
<tr>
<td>I-49</td>
<td>Maximum Noise Y Plus Errors</td>
<td>322</td>
</tr>
<tr>
<td>I-50</td>
<td>Maximum Noise X Position Errors</td>
<td>323</td>
</tr>
<tr>
<td>I-51</td>
<td>Maximum Noise Y Position Errors</td>
<td>324</td>
</tr>
<tr>
<td>I-52</td>
<td>Maximum Noise X Centroid Minus Errors</td>
<td>325</td>
</tr>
<tr>
<td>I-53</td>
<td>Maximum Noise Y Centroid Minus Errors</td>
<td>326</td>
</tr>
<tr>
<td>I-54</td>
<td>Maximum Noise X Centroid Plus Errors</td>
<td>327</td>
</tr>
<tr>
<td>I-55</td>
<td>Maximum Noise Y Centroid Plus Errors</td>
<td>328</td>
</tr>
<tr>
<td>I-56</td>
<td>Maximum Noise X Centroid Position Errors</td>
<td>329</td>
</tr>
<tr>
<td>I-57</td>
<td>Maximum Noise Y Centroid Position Errors</td>
<td>330</td>
</tr>
<tr>
<td>I-58</td>
<td>Maximum Noise Error of Estimated h</td>
<td>331</td>
</tr>
<tr>
<td>I-59</td>
<td>Maximum Noise Error of Estimated Dh/Dx</td>
<td>332</td>
</tr>
<tr>
<td>I-60</td>
<td>Maximum Noise Error of Estimated Dh/Dy</td>
<td>333</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modcomp Tracking and Intensity Function Derivation Errors</td>
</tr>
<tr>
<td>2</td>
<td>Cyber Tracking and Intensity Function Derivation Errors</td>
</tr>
<tr>
<td>3</td>
<td>Modcomp Tracking Errors for Correlator-Kalman Filter Algorithm</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$\hat{x}(t_i^+)$</td>
<td>state estimate vector after measurement incorporation</td>
</tr>
<tr>
<td>$\hat{x}(t_i^-)$</td>
<td>state estimate vector before measurement incorporation</td>
</tr>
<tr>
<td>$K(t_i)$</td>
<td>Kalman filter gain</td>
</tr>
<tr>
<td>$z(t_i)$</td>
<td>measurement vector of average intensities over individual pixel elements (pixels) of the FLIR</td>
</tr>
<tr>
<td>$h(\hat{x}(t_i),t_i)$</td>
<td>intensity shape function for measurements at time $t_i$ as a function of the state estimate</td>
</tr>
<tr>
<td>$x_{\text{centroid}}$</td>
<td>$x$-position of the centroid of the target's intensity profile</td>
</tr>
<tr>
<td>$x_{\text{dynamics}}$</td>
<td>$x$-position of the centroid resulting from target dynamics</td>
</tr>
<tr>
<td>$x_{\text{atmospherics}}$</td>
<td>$x$-position of the centroid resulting from atmospheric jitter</td>
</tr>
<tr>
<td>$\tilde{G}(f_x,f_y)$</td>
<td>Frequency Spectrum of two-dimensional sequence $\check{g}(x,y)$</td>
</tr>
<tr>
<td>$\check{g}(x,y)$</td>
<td>spatial domain 2D presentation</td>
</tr>
<tr>
<td>$f_x,f_y$</td>
<td>spatial frequencies</td>
</tr>
<tr>
<td>$x,y$</td>
<td>space variables</td>
</tr>
<tr>
<td>$\hat{y}(t)$</td>
<td>current averaged value</td>
</tr>
<tr>
<td>$y(t)$</td>
<td>current data frame</td>
</tr>
<tr>
<td>$\hat{y}(t-1)$</td>
<td>previous averaged data frame</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>smoothing constant</td>
</tr>
<tr>
<td>$T_t$</td>
<td>target correlation time</td>
</tr>
</tbody>
</table>
### List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1(t), w_2(t)$</td>
<td>continuous time, independent, white Gaussian noise processes</td>
</tr>
<tr>
<td>$\sigma_d^2$</td>
<td>desired variance on outputs $x_d$ and $y_d$</td>
</tr>
<tr>
<td>$F_t$</td>
<td>truth model plant matrix</td>
</tr>
<tr>
<td>$x_t(t)$</td>
<td>truth model state vector</td>
</tr>
<tr>
<td>$G_t$</td>
<td>truth model noise input matrix</td>
</tr>
<tr>
<td>$w_t$</td>
<td>vector of white Gaussian noise inputs</td>
</tr>
<tr>
<td>$\Phi_t$</td>
<td>truth model state transition matrix</td>
</tr>
<tr>
<td>$Z_{kl}$</td>
<td>output of the $K-l$-th pixel</td>
</tr>
<tr>
<td>$A_p$</td>
<td>area of one pixel</td>
</tr>
<tr>
<td>$I_{max}$</td>
<td>maximum intensity received from target</td>
</tr>
<tr>
<td>$\sigma_g^2$</td>
<td>dispersion of the Gaussian intensity function</td>
</tr>
<tr>
<td>$x_{peak}(t_i), y_{peak}(t_i)$</td>
<td>center of the Gaussian intensity pattern</td>
</tr>
<tr>
<td>$V_{kl}(t_i)$</td>
<td>additive noise for the $K-l$-th pixel corresponding to background and FLIR noises</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>distance in # pixels from pixel $i$ to pixel $j$</td>
</tr>
<tr>
<td>$v'(t_i)$</td>
<td>white Gaussian vector each component zero mean, unit variance, and independent of other components</td>
</tr>
<tr>
<td>$T_f$</td>
<td>correlation time</td>
</tr>
<tr>
<td>$w_f(t)$</td>
<td>white Gaussian noise process</td>
</tr>
<tr>
<td>$\sigma_f$</td>
<td>root mean squared value of $x$</td>
</tr>
<tr>
<td>$F_f$</td>
<td>filter plant matrix</td>
</tr>
<tr>
<td>$x_f(t)$</td>
<td>filter state vector</td>
</tr>
</tbody>
</table>
List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_f(t)$</td>
<td>input white Gaussian noise vector for filter</td>
</tr>
<tr>
<td>$T_{df}$</td>
<td>correlation time assumed for target dynamics</td>
</tr>
<tr>
<td>$T_{af}$</td>
<td>correlation time assumed for atmospheric jitter</td>
</tr>
<tr>
<td>$\sigma_{df}^2$</td>
<td>assumed target dynamics noise variance</td>
</tr>
<tr>
<td>$\sigma_{af}^2$</td>
<td>assumed atmospheric jitter noise variance</td>
</tr>
<tr>
<td>$P_f$</td>
<td>filter state transition matrix</td>
</tr>
<tr>
<td>$P(t_i^+)$</td>
<td>conditional state covariance matrix from measurement update equation at time $t_i$</td>
</tr>
<tr>
<td>$P(t_{i+1}^-)$</td>
<td>conditional state covariance matrix propagated from time $t_i$ to $t_{i+1}$</td>
</tr>
<tr>
<td>$Q_f$</td>
<td>noise covariance Kernel descriptor given in Equation (4-8)</td>
</tr>
<tr>
<td>$H(t_i)$</td>
<td>$\frac{\partial h(\hat{x}(t_i), t_i)}{\partial \hat{x}}$ linearized function of intensity measurements</td>
</tr>
<tr>
<td>$K(x,y)$</td>
<td>output of cross correlation</td>
</tr>
<tr>
<td>$c$</td>
<td>centroid summation</td>
</tr>
<tr>
<td>$Z_c$</td>
<td>estimated x and y coordinates by the correlation/centroid algorithm</td>
</tr>
<tr>
<td>$H_c$</td>
<td>the linear combination of state variables which contribute to the respective measurement elements</td>
</tr>
<tr>
<td>$X_c$</td>
<td>correlator state vector</td>
</tr>
<tr>
<td>$V_c$</td>
<td>additive noise corruption to correlation position estimates</td>
</tr>
<tr>
<td>$M_{eij}$</td>
<td>mean error for pixel $j$ at the time frame $i$</td>
</tr>
<tr>
<td>$\sigma_R$</td>
<td>Cholesky square root</td>
</tr>
</tbody>
</table>

xv
# List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(t_i^-)$</td>
<td>propagated conditional covariance matrix before measurement incorporation at time $t_i$</td>
</tr>
<tr>
<td>$\hat{x}(t_i^-)$</td>
<td>propagated state estimate vector of filter before measurement incorporation at time $t_i$</td>
</tr>
<tr>
<td>$K(t_i)$</td>
<td>Kalman filter gain</td>
</tr>
<tr>
<td>$h(\hat{x}(t_i^-), t_i)$</td>
<td>nonlinear function of intensity measurements at time $t_i$, as function of filter state estimates $\hat{x}(t_i^-)$</td>
</tr>
<tr>
<td>$z(t_i)$</td>
<td>actual realization of measurement vector</td>
</tr>
<tr>
<td>$g(x,y) * l(x,y)$</td>
<td>cross correlation of the two-dimensional spatial sequences $g(x,y)$ and $l(x,y)$</td>
</tr>
<tr>
<td>$L^*(f_x,f_y)$</td>
<td>complex conjugate of the Fourier transform of the sequence $l(x,y)$</td>
</tr>
<tr>
<td>$z_c(t_i)$</td>
<td>the estimated x and y coordinates by the correlation/centroid algorithm</td>
</tr>
<tr>
<td>$H_c$</td>
<td>the linear combination of state variables which contribute to the respective measurement elements</td>
</tr>
<tr>
<td>$X_c$</td>
<td>the state vector</td>
</tr>
<tr>
<td>$V_c(t_i)$</td>
<td>additive noise corruption assumed to be white and Gaussian with statistics to be determined</td>
</tr>
<tr>
<td>$N$</td>
<td>number of simulations</td>
</tr>
<tr>
<td>$k$</td>
<td>index of Monte Carlo simulation runs going from 1 to N</td>
</tr>
<tr>
<td>$h_{tijk}$</td>
<td>value for frame i, pixel j, of the truth model intensity for the kth simulation</td>
</tr>
<tr>
<td>$h_{ijk}$</td>
<td>estimate value for frame i, pixel j, of the intensity as derived in the kth simulation</td>
</tr>
<tr>
<td>$M_{e_i}$</td>
<td>the spatial average (over all the pixels) of the mean pixel errors at the ith frame</td>
</tr>
</tbody>
</table>
### List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>the spatial average of all the ensemble pixel error variances, over all the pixels at the $i$th frame</td>
</tr>
<tr>
<td>$e_i$</td>
<td>variance of the error of $j$th pixel at the $i$th frame</td>
</tr>
<tr>
<td>$E_{x_d_i}$</td>
<td>mean error in $x$ dynamics for frame $i$</td>
</tr>
<tr>
<td>$x_{t_d_i k}$</td>
<td>truth model, $x$ dynamic value at frame $i$ simulation $k$</td>
</tr>
<tr>
<td>$\hat{x}_{d_i k}$</td>
<td>filter estimated $x$ dynamic value at frame $i$ simulation $k$</td>
</tr>
<tr>
<td>$K_1$</td>
<td>complex constant</td>
</tr>
<tr>
<td>$T_0$</td>
<td>Fourier transform of $t_0(x_1, y_1)$</td>
</tr>
<tr>
<td>$\tau$</td>
<td>mean wavelength of the laser</td>
</tr>
<tr>
<td>$a, k_0$</td>
<td>constants</td>
</tr>
<tr>
<td>$\hat{L}(t_{i_1})$</td>
<td>current averaged data frame in frequency domain</td>
</tr>
<tr>
<td>$L(t_{i_1})$</td>
<td>transform of current noise corruption data frame</td>
</tr>
<tr>
<td>$\hat{L}(t_{i-1})$</td>
<td>previous result of averaging in frequency domain</td>
</tr>
</tbody>
</table>
Abstract

Considerable work has been accomplished at AFIT in the last three years to improve the tracking capability of the high energy laser weapon. The improvements were achieved via use of an adaptive extended Kalman filter algorithm. In this research, work is initiated on a tracker able to handle "multiple hot spot" targets, in which digital signal processing is employed on the FLIR data to identify the underlying target shape. This identified shape is then used in the measurement model portion of the filter as it estimates target offset from the center of the field-of-view. Two tracking algorithms are developed. The first algorithm uses an extended Kalman filter to process the intensity measurements from a FLIR to produce target position estimates. The second algorithm uses a linear Kalman filter to process the position estimates of an improved correlation algorithm. This algorithm is improved over standard correlators by using thresholding to eliminate poor correlation information, dynamic information from the Kalman filter and it also uses the on-line derived target shape.
I. Introduction

The first successful laser is generally credited to Maiman at the Hughes Research Laboratories in 1960. Since that time, there has been an explosive growth in laser technology. High energy lasers have become prime candidates for use as weapon systems (1:87), and in fact have already been tested against airborne targets (2:14-17; 3:16-19). The use of high energy lasers for military applications may have a significant impact on any future war (1:87).

Laser weapon systems are highly desirable because of their potential to deposit large amounts of energy on a small area of a target in a short period of time. However, there are many technological problems which must be resolved prior to high energy lasers being used as efficient and effective weapon systems.

The problem of very accurate target position estimation, for tracking the target, is one such technical problem. An Infra-Red (IR) laser beam must be kept positioned on a specific part of a target for an extended amount of time to destroy a target or critical components of a target. An aiming angle correct to within "six one hundred thousandths of one degree", .1 microrad, "is required to hit a missile 5000 Km away, which
is a precision beyond existing technology for both the U.S. and the U.S.S.R." (1:87-89).

**Background**

This research investigates potential solutions to the target tracking problems associated with directed energy beam weapons. The Air Force Weapons Laboratory (AFWL) located at Kirtland Air Force Base, New Mexico, currently uses correlation trackers to provide precise target position estimates to feedback controllers in the presence of several disturbances. These disturbances include any effect that can cause relative motion between the beam and the target, such as target motion, mirror vibration, atmospheric jitter, and sensor measurement errors (4:2). A correlation tracker stores a complete set of predetermined or previous real-time data which provide a template to compare with new information to be received at a later time. Cross correlation techniques are used to estimate the relative position offsets from one frame of data to the next. This relative position information is then used to control the gimbals which keep the target centered within the sensor's field-of-view. One possible sensor, the one currently of most interest, is the Forward Looking Infra-Red sensor (FLIR).

The high energy laser is servoed to the FLIR so that centering the target within the FLIR's field-of-view physically points the laser toward the target. This type of
tracker needs no prior information about the target's shape or dynamics and is therefore well suited to many general applications. The disadvantages of this tracker are its susceptibility to noise and its insensitivity to any knowledge of the target's dynamics.

To overcome these specific deficiencies of the correlator tracker, the use of an extended Kalman filter tracker has been investigated (4;5;6). Appendix J contains a very brief generic view of the Kalman filter equations and then provides information on how they are used for this application. In many practical tracking problems, the type of target being tracked is known along with target parameters such as size, shape and even acceleration characteristics. Additional information incorporated by the Kalman filter, which is not utilized by correlation trackers, is knowledge of the statistical effects of atmospheric distortion on the radiated wavefront as it propagates to the FLIR. The Kalman filter uses this information to aid in separating the true target relative motion from the atmospheric disturbance. The Kalman filter operating upon the raw digitized image has been shown to perform well in comparison to correlator trackers when the target intensity function shape is relatively well known and unimodal, and when the internal filter models depict the actual tracking situation reasonably well (5;6). These studies found the extended Kalman filter tracker to be a
robust tracker, even in low signal-to-noise-ratio environments.

Problem

A major difficulty in using an extended Kalman filter for enhanced IR tracking is the Kalman filter algorithm's requirement for an accurate reference image, \( h(\mathbf{x}(t_i), t_i) \) and the derivatives, \( \partial h(\mathbf{x}(t_i), t_i) / \partial \mathbf{x} = \mathbf{H}(\mathbf{x}(t_i), t_i) \) of that reference image with respect to the states, over the entire FLIR image. This research adopts the notation that an underlined lower case letter denotes a vector while an uppercase letter underlined denotes a matrix. In previous research efforts, this reference image was assumed to be a known Gaussian functional form. During the past year, work was initiated on a tracker able to handle "multiple hot spot" targets, in which digital processing must be employed to identify the target's shape (4). This identified shape could then be used in the measurement model portion of the Kalman filter as it estimates target offsets from the center of the field-of-view. The state estimates, in turn, are incorporated into the shape function determination. The data processing algorithm is shown in Figure 1. The Kalman filter processes the measurement vector \( z(t_i) \) using the equation

\[
\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i) [z(t_i) - h(\hat{x}(t_i^-), t_i)]
\]

where
\( \hat{x}(t_i^+) \) = state estimate vector after measurement incorporation at time \( t_i \)

\( \hat{x}(t_i^-) \) = state estimate vector propagated from previous measurement update to time \( t_i \)

\( K(t_i) \) = Kalman filter gain

\( z(t_i) \) = Measurement vector of average intensities over individual picture elements (pixels) of the FLIR array

\( h(\hat{x}(t_i^-), t_i) \) = intensity shape function for measurements at the time \( t_i \) as a function of the state estimate

The apparent location of the target within the sensor's field-of-view is the sum of effects due to true target dynamics and atmospheric disturbances. The four-state estimate vector, \( \hat{x}(t_i) \), consists of estimates for the \( x \) and \( y \) positions due to true target dynamics and the \( x \) and \( y \) positions due to atmospherics.

Figure 1 shows two data processing paths for the intensity measurements. The FLIR is positioned so that the center of its field-of-view is where the Kalman filter predicted the true target position to be, resulting from dynamics over the most recent sample period. Each FLIR frame is arranged by rows into a measurement vector \( z(t_i) \) which is the input to the Kalman filter in the lower path of Figure 1. The Kalman filter uses the nonlinear intensity function evaluated at the predicted state, \( h(\hat{x}(t_i^-), t_i) \), and the linearized intensity function, \( h(\hat{x}(t_i^-), t_i) \), for that frame to compute an updated state estimate, \( \hat{x}(t_i^+) \), from the measurement vector \( z(t_i) \).
Figure 1. Data Processing Algorithm
The Kalman filter then uses its internal dynamics model to propagate its state estimates forward to produce its best estimate of the states at the next sample time, $\hat{x}(t_{i+1})$. That information is passed to the controller, which will position the FLIR so that the center of its field-of-view is where the filter predicts the dynamics will place the target position at that next sample time. The upper path of Figure 1 is designed to provide the linearized and nonlinear intensity function for use by the Kalman filter.

The nonlinear intensity pattern, $h(\hat{x}(t_i^-), t_i)$, is the noise-free intensity measurement frame expected if the Kalman filter's propagated state estimates were perfect. Since the center of the field-of-view is equal to the filter-predicted target location due to the controller action, the nonlinear $h[\hat{x}(t_i^-), t_i]$ will be the target's intensity profile, $h$, offset from the center of the FLIR's field-of-view by the predicted atmospheric states. This is because the x-axis position of the centroid of the target's intensity profile is defined by

$$x_{\text{centroid}} = x_{\text{dynamics}} + x_{\text{atmospherics}} \quad (1-2)$$

and the control action is specifically intended to zero out the estimated $x_{\text{dynamics}}$, and similarly for the y-axis. The linearized intensity pattern, $H[\hat{x}(t_i^-), t_i]$, is also used by the Kalman filter, within the computation of the gain matrix $K(t_i)$, to incorporate a measurement to produce an updated
state estimate, \( \hat{x}(t_{i+}) \). This linearized intensity pattern is the derivative of the nonlinear intensity function, \( h(\hat{x}(t_{i-}), t_i) \), with respect to a change in the Kalman filter states evaluated at the predicted state at that sample time, \( \hat{x}(t_{i-}) \):

\[
\frac{\partial h(\hat{x}(t_{i-}), t_i)}{\partial x} = \frac{\partial h(\hat{x}(t_{i-}), t_i)}{\partial x}.
\]

To generate these estimated target intensity patterns from the noise-corrupted FLIR requires interframe filtering to attenuate the noise. This interframe smoothing requires the target's intensity profile to be centered from frame to frame so that the noise can be averaged out. To center the target's intensity function within a data array which represents a FLIR's field-of-view, the shifting theorem of the Fourier Transform is used.

The offset of the target's intensity pattern from the center of the FLIR's field-of-view can be negated. This is accomplished by multiplying the Fourier Transform of the frame of data by the complex conjugate of the linear phase shift induced in the frequency domain by the translation of the intensity pattern from the center of the frame in the space domain. Atmospheric jitter and imperfect propagation of dynamic states are the sources of the image's translation from the center of the sensor's field-of-view in the space domain. The output of the Kalman filter's update Equation (1-1) provides an estimate of both of these offsets for the present frame of data, \( \hat{x}(t_{i+}) \). These estimates of the updated
target location and atmospheric jitter, $\hat{x}(t_i^+)$, are used to compute the estimated centroid location, as shown by Equation (1-2). The centroid location is used in the argument of the complex conjugate of the linear phase shift which negates the spatial translation effects and provides a centered intensity function. This centered pattern is then averaged with previously centered frames of data, using exponential smoothing, to reduce the effects of noise corruption. The derivative property of the Fourier Transform is then used to differentiate the centered smoothed intensity function with respect to a change in Kalman filter states.

This shape function and its derivatives are then evaluated at the state expected at the next sample time. For this application, where the FLIR is centered on the predicted target location, the intensity patterns are evaluated at the predicted atmospheric states. This is the intensity pattern which would be received if the image were noise-free and the Kalman filter had made perfect estimates of the dynamic and atmospheric states, of Equation (1-2), which determine the location of the centroid of the intensity pattern. The inverse Fourier transform is then computed and $h[\hat{x}(t_{i+1}^-), t_{i+1}]$ and $H[\hat{x}(t_{i+1}^-), t_{i+1}]$ are ready for the Kalman filter to process the next frame of data.

Plan of Attack

This section provides an overview of the general flow.
of this research, as pictured schematically in Figure 2.

![Research Plan Flow Diagram](image)

**Figure 2. Research Plan of Attack Flow Diagram**

The first block of Figure 2 represents the first step of this research. This step includes the development of the algorithms necessary to process the FLIR measurements. The complete implementation of Figure 1 was accomplished during this portion of the research. This included the development of the four-state extended Kalman filter needed to estimate
target position along with the estimated atmospheric disturbance in the x and y directions. To generate the intensity functions \( h(\hat{x}(t_i^-), t_i) \) and \( H(\hat{x}(t_i^-), t_i) \), algorithms were needed to take the forward/inverse two-dimensional Discrete Fourier Transform of the FLIR data. Once in the frequency domain, the capability to shift the data to center it within the frame was developed so interframe filtering could be accomplished. An exponential smoothing algorithm was developed for filtering out background noise. The derivative of this smoothed intensity function was derived via the derivative property of the Fourier transform. The shifting algorithm is then used to compute what the intensity function and its derivative would be as evaluated at the filter-predicted state at the next sample time.

The next objective of this research was to perform an analysis of the target tracking (and shape function generation secondarily) performance and stability of the data processing algorithm of Figure 1. Monte Carlo analysis techniques were used to gather statistics on the ability of this data processing algorithm to generate \( h(\hat{x}(t_i^-), t_i) \) and \( H(\hat{x}(t_i^-), t_i) \). It is the target tracking performance which is of primary concern.

The alternate data processing algorithm evaluation section evaluates the potential of an improved correlator tracker followed by a Kalman filter. The correlator tracker
works well in high signal-to-noise ratio scenarios and provides good estimates of centroid location, especially for extended targets (as opposed to point source targets). The Kalman filter could then filter these estimates, treated as measurements for the filter, to separate out the components of Equation 1-2.

The demand for real-time operation of the computationally intense algorithms discussed above requires the development of parallel processing techniques such as optical processing. The final section is a study into the potential of optically computing some of the quantities of Figure 1.

Overview

Chapters II, III, and IV will describe in detail the mathematical models used in the computer implementation. Chapter II presents the algorithms used in the derivation of the nonlinear and linearized intensity functions. Chapter III presents the mathematical model used for the truth model which represents the environment from which measurements are taken. Chapter IV describes the Kalman filter model used in the computer simulation. Chapter V discusses the possibility of a correlator followed by a Kalman filter to provide the position estimates. Chapter VI presents the results from the performance analysis which includes the statistics on how well the intensity functions are being constructed as well as statistics on target tracking ability for both
algorithms, and Chapter VII presents the optical processing options. Finally, Chapter VIII will present the conclusions and recommendations.
II. Derivation of the Nonlinear and Linearized Intensity Functions

Introduction

This chapter will present the algorithms necessary to produce an accurate reference intensity profile, \( h[\hat{x}(t_i^-), t_i] \), and the derivatives of that function, \( H[\hat{x}(t_i^-), t_i] \), with respect to the states. These intensity functions are needed by the extended Kalman filter to process FLIR measurements to update state estimates (see Chapter IV, Extended Kalman Filter). The true intensity profile is a vector of scalar components equal to the average intensity of the target's IR image over a particular pixel in the field-of-view. Noise-corrupted FLIR data must be processed to determine the target's profile. Figure 1 showed the data processing necessary to produce the intensity functions from the incoming data. The goal is to recognize the intensity function which is common to the noise-corrupted frames of data.

Pattern Recognition

All of the information of a two-dimensional intensity pattern can be preserved by a set of eigenvalues and eigenfunctions. To retain all of the information of a profile may require an infinite set of such values and functions. It is desirable to examine the FLIR image data in a coordinate system which has properties more conducive to recognizing patterns than the spatial x-y coordinate system. Ideally, a
linear transformation which rotates the FLIR image into a new coordinate space where the components are uncorrelated is desired. The Karhunen-Loeve Transformation is an example of such an operation. This transformation does possess the disadvantage of having to produce the correlation matrix which is $N^2 \times N^2$ if the input square matrix is $N \times N$.

The Karhunen-Loeve Transformation possesses the property of providing a coordinate space with perfectly uncorrelated image components, but it is a difficult transformation to perform in its exact form. The Karhunen-Loeve Transformation does provide motivation for the use of the more familiar Fourier Transform which uses complex exponentials as eigenfunctions (4:12). Where, in the Karhunen-Loeve Transformation the eigenvalues correspond to actual variance statistics projected on coordinate axes of basis vectors in a space where the image data is uncorrelated, the Fourier Transform is a projection of the image along the basis vectors formed by complex exponentials (9:195).

Although the Fourier Transform does not provide the perfect decorrelation, it does possess a computational advantage which outweighs the compression efficiency of the Karhunen-Loeve Transformation (9:194). The Fourier Transform also possesses the advantage that the two-dimensional transform is separable and can be obtained by one-dimensional operations.
Two-Dimensional Fourier Transforms

The purpose of decomposing the information contained within the complicated image intensity data into a set of eigenvalues and eigenfunctions is to compress that information into a set of inputs which will make manipulations performed on the data easier. Fourier analysis provides one way of performing such a decomposition.

The Fourier Transform of a complex-valued function \( g(x,y) \) of two independent variables is a decomposition into a linear combination of elementary functions of the complex exponential form \( \exp[j2\pi(f_x x + f_y y)] \) \((7:8)\). This Fourier Transform is defined by

\[
\mathcal{F}(g(x,y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) \exp[-j2\pi(f_x x + f_y y)] dx dy \tag{2-1}
\]

where

- \( \mathcal{G}(f_x, f_y) \) = Frequency Spectrum
- \( g(x,y) \) = Spatial Domain Representation
- \( f_x, f_y \) = Spatial Frequencies
- \( x, y \) = Spatial Variables

The transform is also a complex-valued function of two independent variables \( f_x \) and \( f_y \), which are the spatial frequencies. The inverse Fourier Transform is defined by

\[
\tilde{g}(x,y) = \mathcal{F}^{-1}(\mathcal{G}(f_x, f_y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{G}(f_x, f_y) \exp[j2\pi(f_x x + f_y y)] df_x df_y \tag{2-2}
\]

The complex-valued number \( \mathcal{G}(f_x, f_y) \) is a weighting factor that...
is applied to the eigenfunction, which represents the coordinate axes of the Fourier domain, in order to synthesize \( \tilde{g}(x,y) \).

The Fourier Transform is separable, so that the two-dimensional transform domain can be reached with one-dimensional operations. Implementation is accomplished by transforming first on the rows of the image, followed by the transformation along the columns.

The two-dimensional finite Discrete Fourier Transform of a two-dimensional periodic sequence, of period \( N \) in both directions, is the finite sum of complex exponentials in the form

\[
\tilde{G}(f_x,f_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{g}(x,y) \exp[-j \frac{2\pi}{N}(xf_x + yf_y)] \quad (2-3)
\]

The complex sequence of intensity values, \( \tilde{g}(x,y) \) is discretized into an \( N \)-pixel by \( N \)-pixel array. The inverse Discrete Fourier Transform is defined by

\[
\tilde{g}(x,y) = \frac{1}{N^2} \sum_{f_x=0}^{N-1} \sum_{f_y=0}^{N-1} \tilde{G}(f_x,f_y) \exp[j \frac{2\pi}{N}(f_x x + f_y y)] \quad (2-4)
\]

The two-dimensional complex-valued \( N \times N \) array, \( \tilde{G}(f_x,f_y) \), which represents the two-dimensional finite Discrete Fourier Transform of the original FLIR data, is discretized into spatial frequency bins. Each component of the array represents one of the \( N/2 \) spatial frequencies or their conjugates. The inverse transform differs from the forward transform only in
the sign of the exponent which appears in the complex exponential within the summation of Equation (2-4). The transform, $\tilde{G}(f_x, f_y)$, will be periodic of period $N$ as was the original sequence, $\tilde{g}(x, y)$. Appendix A contains a further interpretation of the two-dimensional Discrete Fourier Transform. It also contains details on how the two-dimensional Fourier Transform implementation was tested.

In summary, the two-dimensional Discrete Fourier Transform assumes a finite length two-dimensional sequence is one period of a two-dimensional periodic sequence which then can be decomposed into a linear combination of complex exponentials. See Appendix A for explicit details.

**Shifting Property of Fourier Transforms**

To generate the target intensity patterns from the noise-corrupted FLIR data, interframe filtering is required to attenuate the noise. Interframe smoothing requires that the target's intensity profile be centered from frame to frame. Successive centered frames of data can then be averaged to attenuate the corrupting noise. The shifting property of the Fourier Transform is used in conjunction with the filter's estimated location of the intensity profile to center the data.

A shift in the scalar spatial domain of a finite duration one-dimensional sequence can be interpreted as a rotation in the basic interval (9:103). That is, the shifting of
samples out of one side of the periodic sequence results in identical samples entering the interval at the other end. This is sometimes called a cylindrical shift. If, for example, the Fourier Transform of \( \tilde{g}(x) \) is defined by

\[ F[\tilde{g}(x)] = \tilde{G}(\omega) \quad (2-5) \]

then the Fourier Transform of the shifted sequence is defined by

\[ F[\tilde{g}(x-x_0)] = e^{-j\omega x_0} \tilde{G}(\omega) \quad (2-6) \]

In the case of a finite-area two-dimensional sequence, a translation of the function in the space domain introduces a linear phase shift in the frequency domain (9:9):

\[ F[\tilde{g}(x-x_0, y-y_0)] = \tilde{G}(f_x, f_y) \exp(-j2\pi(f_x x_0 + f_y y_0) \quad (2-7) \]

where \( F[\tilde{g}(x,y)] = \tilde{G}(f_x, f_y) \)

An interpretation of the translation by \( x_0 \) and \( y_0 \) in the space domain, analogous to the one-dimensional interpretation, would be a rotation of each column by \( x_0 \) samples and each row by \( y_0 \) samples (9:119).

The offset of the target's intensity pattern from the center of the FLIR's field-of-view can be negated to provide a centered image for smoothing. This is accomplished by multiplying the Fourier Transform of the frame of data, which contains the offset intensity profile, by the complex conjugate of the linear phase shift induced in the frequency domain. The centered intensity profile can be obtained from
the offset profile by

\[
\tilde{g}(x,y) = F^{-1}\{F[\tilde{g}(x-x_0, y-y_0)] \exp[+j2\pi(f_x x_0 + f_y y_0)]\}
\]  

(2-8)

In summary, by using the Kalman filter's updated estimates of the location of the centroid within a frame of data, the centered intensity profile can be obtained. Appendix B contains the details of how the shift is implemented.

Smoothing in the Fourier Domain

The target's intensity profile is not directly available for observation. This is because each FLIR frame is corrupted by inherent FLIR errors and background noise. Background noise can vary from zero to high temporal and spatial correlations. The FLIR errors, such as thermal noise and dark current effects, can be modelled as temporally and spatially uncorrelated noise. The effects of these noises on target intensity profile shape generation must be minimized by some appropriate data processing techniques. Any chosen data processing technique must not assume the precise form of the corrupting noise because of the large range of correlation characteristics. However, it is appropriate to exploit the fact that the corrupting noise changes faster from sample period to sample period than the target's intensity pattern (4;5).

An exponential smoothing algorithm was chosen to combat the effects of the corrupting noise. The equation which
defines exponential smoothing is:

\[ \hat{y}(t) = \alpha y(t) + (1-\alpha) \hat{y}(t-1) \quad (10.115) \] (2-9)

where

\[ \hat{y}(t) = \text{current averaged value} \]
\[ y(t) = \text{current data frame} \]
\[ \hat{y}(t-1) = \text{previous averaged data frame} \]
\[ \alpha = \text{smoothing constant } 0 < \alpha < 1 \]

The smoothing constant \( \alpha \) can vary anywhere from 0 to 1. The value of \( \alpha \) determines how the current averaged data frame, \( \hat{y}(t) \), responds to the current data frame, \( y(t) \). If the target's intensity pattern is only changing slowly, the value of \( \alpha \) should tend more toward zero than one so that more emphasis is placed on the previous averaged data but the present frame is still used.

In summary, \( \hat{y}(t) \) contains information from all the past data frames. The initial data frames have less of an effect on the most recent averaged data frame, \( \hat{y}(t) \), than do the most recent data, as is appropriate for a slowly changing target image. Appendix C contains the details of how this smoothing algorithm was implemented and how the algorithm can be expressed in a closed form.

**Derivative Property of Fourier Transforms**

As stated in Chapter I and to be shown explicitly in Chapter IV, the extended Kalman filter algorithm requires
the derivatives of the intensity function with respect to the states, $H[x(t_i^-), t_i] = \frac{\partial h(x(t_i^-), t_i)}{\partial x}$. In past research efforts (4:), this linearized intensity function was derived using a numerical approximation known as the Forward-Backward Difference Method. This method only uses the pixels just before and just after the pixel being computed in the direction the spatial derivative is being taken. It was for this reason that the Derivative Property of the Fourier Transform, which uses all the data array, was chosen as a better means of generation for this research.

Differentiation in the space domain just becomes a multiplication by $j\pi (f_x + f_y$) in the frequency domain. The derivative of the intensity function can be expressed in terms of the transform of the function

\[
F \left[ \frac{\partial h(x,y)}{\partial x} \right] = j\pi f_x \cdot F[h(x,y)] \tag{2-10a}
\]

\[
F \left[ \frac{\partial h(x,y)}{\partial y} \right] = j\pi f_y \cdot F[h(x,y)] \tag{2-10b}
\]

Appendix D provides explicit details on the implementation of this algorithm.

In summary, the derivatives of the intensity function with respect to the states are required by the extended Kalman filter algorithm. These derivatives can be expressed in terms of the transform of that intensity function using Equations (2-10a) and (2-10b). Appendix D contains the details of Subroutine Deriv.
Derivation of Intensity Functions: Summary

This section will review the algorithms used to produce an accurate reference intensity profile, $h(\hat{x}(t_i^-), t_i)$, and the derivatives of that function $H(\hat{x}(t_i^-), t_i)$. As stated in Chapter I, the extended Kalman filter algorithm requires these intensity functions for the update state equations, (see Chapter IV, Extended Kalman Filter).

Noise-corrupted FLIR data must be processed to determine the target's intensity profile. The upper path of Figure 1 showed the data processing necessary to produce the intensity functions from the incoming noise-corrupted FLIR data. The goal of the data processing is to recognize the intensity function which is common to the noise-corrupted frames of data.

Interframe smoothing is required to generate the estimated target intensity pattern from the FLIR data. This smoothing is accomplished in the frequency domain since that is where the shift is accomplished, so the two-dimensional transform is first taken of the data. It will be shown in Appendix C on exponential smoothing that smoothing in the space domain is equivalent to smoothing in the frequency domain (see Table I, Chapter VI). To average from frame to frame, the target's intensity profile must be centered. This is accomplished via use of the Shifting Theorem of the Fourier Transform. Once the data is centered, exponential
smoothing is used to attenuate the noise while still in the frequency domain. The spatial derivatives are then obtained from this smoothed data by using the derivative property of the Fourier Transform. The shifting property can then be used again to derive the intensity functions evaluated at the predicted states. The high energy laser will be pointed at the best estimate of the target's position as calculated by the Kalman filter. The target's intensity function will be offset by atmospheric jitter from that target location. Since the field-of-view is centered at the estimated target location, the algorithm expects the intensity function to be offset from the center of the field-of-view by the estimated atmospheric jitter. By using the shifting property the estimated centered target intensity functions can be manipulated to produce the expected offset intensity functions.

In summary, this chapter presented the algorithms used to produce the intensity functions needed by the extended Kalman filter algorithm. The use of this procedure eliminates the need to make assumptions on the target's intensity profile.
III. Truth Model

Introduction

The truth model is the best mathematical representation of the real world environment from which the measurement vector, \( z(t_i) \), is generated. The environmental characteristics which are modelled include the underlying target dynamics, the atmospheric jitter effects, and the background and FLIR noises. This chapter presents the models used, along with an explanation of how the measurement vector, \( z(t_i) \), was created. The first two sections of this chapter present the two processes which contribute to movement of the intensity function, target dynamics and atmospheric jitter. These models are then transformed into state space notation and the propagation equations are given. Information on the computer creation of the measurement data for the computer simulation is next, with the final section containing the model for spatial correlations of background noise.

Target Dynamics Model

For this research, both deterministic and stochastic dynamic models were used. In the deterministic model, a target moving across the image plane at a constant velocity was simulated. In the stochastic model, a first order Gauss-Markov process was used to describe the movement of the target with respect to the field-of-view of the sensor. Although
there exist better models for describing the movement of
specific targets, the very general stochastic model is applic-
cable to many general targets. Together, these two models
yield some of the basic characteristics of trajectories.

The deterministic model was developed to test the algo-

\[
\begin{align*}
\bar{x}_d(t_{i+1}) &= x_d(t_i) + .15 \\
\bar{y}_d(t_{i+1}) &= y_d(t_i) + .15
\end{align*}
\] (3-1)

rithm of Figure 1 with a target which is maintaining a con-
stant velocity across the FLIR sensor's image plane. In this
project, a constant velocity of three pixels over any twenty-
frame time history was used.

The stochastic model developed in Mercier's thesis,
(5:9-10), for target dynamics was also used. In this model,
the output of a first order linear shaping filter driven by
zero mean white Gaussian noise was used to describe the move-
ment of the target with respect to the FLIR's field-of-view.
The outputs of these first order shaping filters, which de-
scribe the target dynamics in each direction, can be expressed
as

\[
\begin{align*}
\dot{x}_d(t) &= \frac{1}{T_t} x_d(t) + w_1(t) \\
\dot{y}_d(t) &= \frac{1}{T_t} y_d(t) + w_2(t)
\end{align*}
\] (3-2)

where

\[
E[w_1(t)] = E[w_2(t)] = 0
\] (3-4)
\[
E[w_1(t)w_1(s)] = E[w_2(t)w_2(s)] = \frac{2\sigma_d^2}{T_t^2} \delta(t-s) \tag{3-5}
\]

\[
E[w_1(t)w_2(s)] = 0 \tag{3-6}
\]

and

\[T_t = \text{target correlation time}\]

\[w_1(t), w_2(t) = \text{continuous time, independent, white Gaussian noise processes}\]

\[\sigma_d^2 = \text{desired variance on outputs } x_d \text{ and } y_d\]

**Atmospheric Jitter Model**

This section presents the mathematical model of the translational position changes of the target's intensity distribution due to disturbances in the atmosphere causing phase distortions in the radiated wavefronts. The model used is based on a study accomplished by The Analytic Sciences Corporation (11:29-30) and data analysis by Hogge and Butts (12:). As a part of these studies, a third order shaping filter driven by white Gaussian noise was developed to generate an output with a power spectral density representation that well approximated that of the effects of atmospheric turbulence. The jitter of the target intensity in each independent direction is modelled as the output of the third order shaping filter shown in Figure 3 (5:12). This filter has a single pole at 14.14 rad/sec and a double pole at 659.5 rad/sec, and it is driven by white Gaussian noise that is zero mean and has a strength of unity:
where

\[ w = \text{zero mean, unit strength, white Gaussian noise process} \]

\[ \omega_1, \omega_2 = \text{break frequencies (}\omega_1 = 14.14 \text{ rad/sec,} \]
\[ \omega_2 = 659.5 \text{ rad/sec} ) \]

\[ y = \text{output of system} \]

\[ K = \text{system gain} \]

Figure 3. Third Order Shaping Filter
\[ E[w(t)] = 0 \] \hspace{1cm} (3-7)

\[ E[w(t) w(s)] = 16(t-s) \] \hspace{1cm} (3-8)

As shown by Mercier, (5), by adjusting the value of the system gain, \( K \), the desired root mean squared (RMS) atmospheric jitter characteristic is achieved. Appendix C of Mercier's thesis contains the details of this derivation.

**State Space Model**

The mathematical models developed in this chapter will now be transformed into state space notation. This transformation results in a time invariant vector differential equation of the form

\[ \dot{x}_t(t) = F_t x_t(t) + G_t w_t(t) \] \hspace{1cm} (3-9)

where

\[ F_t = \text{truth model plant matrix} \]

\[ x_t(t) = \text{truth model state vector} \]

\[ G_t = \text{truth model noise input matrix} \]

\[ w_t = \text{vector of white Gaussian noise inputs} \]

\[ E[w_t(t)] = 0 \] \hspace{1cm} (3-10)

\[ E[w_t(t)w_t^T(t + \tau)] = Q_t \delta(\tau) \] \hspace{1cm} (3-11)

The truth model state vector is defined by
If the Jordan canonical form, as developed by Mercier, is used, the plant matrix of Equation (3-9) becomes

\[
X_t(t) = \begin{bmatrix}
    x_d(t) \\
    x_{1a}(t) \\
    x_{2a}(t) \\
    x_{3a}(t) \\
    y_d(t) \\
    y_{1a}(t) \\
    y_{2a}(t) \\
    y_{3a}(t)
\end{bmatrix}
\]  

(3-12)

\[
P_t = \begin{bmatrix}
    \frac{-1}{T_t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & -\omega_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & -\omega_2 & 1 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & -\omega_2 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & \frac{-1}{T_t} & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & -\omega_1 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 & -\omega_2 & 1 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 & 0 & -\omega_2 & -\omega_2
\end{bmatrix}
\]

\[
\omega_1 = 14.14 \text{ rad/sec} \quad \omega_2 = 659.5 \text{ rad/sec} \quad (3-13)
\]

The truth model noise matrix (5:11-14) is defined by
Using Equations (3-5) and (3-8), $Q_t$ of Equation (3-11) becomes

$$Q_t = \begin{bmatrix}
\frac{2\sigma_d^2}{T_t} & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \frac{2\sigma_d^2}{T_t} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

(3-15)

The movement of the target's intensity profile, the centroid, is governed by components of these truth model matrices where

$$x_{\text{peak}}(t) = x_d(t) + x_a(t)$$

$$y_{\text{peak}}(t) = y_d(t) + y_a(t)$$

(3-16a)
Thus, the location of the centroid of the intensity profile can be expressed as

\[
y_p(t) = \begin{bmatrix} \gamma_{\text{peak}}(t) \\ \gamma_{\text{peak}}(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} x(t) \]

(3-17)

**Propagation Equations**

Propagation equations describe how the state vector makes the transition from one sample time to another. The solution to Equation (3-9) (5:14) is

\[
x(t+1) = \Phi_t(t_1+t) x(t_1) + \int_{t_i}^{t_i+1} \Phi_t(t_1+t) \mathcal{G}_t(t) y(t) \, dt
\]

(3-18)

where \( \Phi_t(t_1+t) \) is the solution to the matrix differential equation

\[
\dot{\Phi}_t(t_1+t) = \mathcal{F}_t \Phi_t(t_1+t)
\]

(3-19a)

and initial condition

\[
\Phi_t(t_1,t_1) = \mathcal{I} \text{ (identity matrix)}
\]

(3-19b)

The matrix \( \Phi_t(t_1,t_1) \) is the state transition matrix which describes the truth model state's transition from time \( t_1 \) to
time $t_{i+1}$. The truth model plant matrix, $E_t$ of Equation (3-14), is a constant matrix, and therefore, the state transition matrix is only a function of the time difference $(t_{i+1} - t_i)$. Therefore, the state transition matrix becomes a constant for a fixed sample time $\Delta t = t_{i+1} - t_i$. The independent channels reflected in the truth model plant matrix result in the state transition matrix being block diagonal with both of the 4-by-4 diagonal blocks equal to

$$
\Phi_{tx}(t_{i+1}, t_i) = \Phi_{ty}(t_{i+1}, t_i) = \begin{bmatrix}
e^{-\Delta t/T_t} & 0 & 0 & 0 \\
0 & e^{-\omega_1 \Delta t} & 0 & 0 \\
0 & 0 & e^{-\omega_2 \Delta t} & \Delta t e^{-\omega_2 \Delta t} \\
0 & 0 & 0 & e^{-\omega_2 \Delta t}
\end{bmatrix}
$$

where $\Delta t =$ constant sampling time $(t_{i+1} - t_i)$. A probabilistic interpretation of the integral term of Equation (3-18) is required to describe the process $x_t(t)$. This integral has a zero mean Gaussian distribution. For notational convenience the contribution of the integral is set equal to $w_{td}(t_i)$.

$$
w_{td}(t_i) \triangleq \int_{t_i}^{t_{i+1}} \Phi_t(t_{i+1}, \tau) G_t(\tau) w_t(\tau) d\tau \quad (3-21)
$$

Computing the statistics of $w_{td}(t_i)$ (5:15) results in

$$
E[w_{td}(t_i)] = 0 \quad (3-22)
$$
The equivalent discrete-time model (5:15) for the above equations is given by Mercier to be

\[ x(t_{i+1}) = \Phi(t_{i+1}, t_i) x(t_i) + \sqrt{Q}_{td} w(t_i) \]  

(3-25)

where \( \sqrt{Q}_{td} \) is the Cholesky square root (13:370) of

\[ Q(t_{i+1}, t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) G(t, \tau) G^T(t, \tau) \Phi^T(t_{i+1}, \tau) d\tau \]  

(3-26)

and

\[ E[w(t_i)] = 0 \]  

(3-27)

\[ E[w(t_i) w^T(t_j)] = \delta_{ij} \]  

(3-28)

The lower triangular Cholesky square root of \( Q_{td} \) satisfies the equation

\[ \sqrt{Q}_{td} \sqrt{Q}_{td}^T = Q_{td} \]  

(3-29)

and can be uniquely defined for a given \( Q_{td} \).

**Measurements**

A measurement history of FLIR data frames is used by the Kalman filter to predict the pointing direction for the high energy laser. The output of an arbitrary pixel, say the \( k-l \)-th pixel, within the FLIR data frame is the average
IR intensity over that pixel as sensed by a detector, and is described mathematically as

\[
z_{kl}(t_i) = \frac{1}{A_p} \int \int I_{\text{target}}(x, y, x_{\text{peak}}(t_i), y_{\text{peak}}(t_i)) \, dx \, dy + V_{kl}(t_i)
\]

where

\[I_{\text{target}}=\text{the intensity of the target as a function of the location in the frame and the centroid peak as defined by Equation (1-2)}\]

\[z_{kl}(t_i) = \text{the output of the k-\ell-th pixel at a time } t_i\]

\[A_p = \text{the area of one pixel}\]

\[(x, y) = \text{coordinates of any point in the k-\ell-th pixel}\]

\[(x_{\text{peak}}(t_i), y_{\text{peak}}(t_i)) = \text{the center of the Gaussian intensity pattern}\]

\[V_{kl}(t_i) = \text{the additive noise for the k-\ell-th pixel corresponding to background and FLIR noises.}\]

in the general case. Using a bivariate Gaussian target intensity profile with circular equal intensity contours as a special case (4:25), the mathematical description becomes

\[
z_{kl}(t_i) = \frac{1}{A_p} \int \int I_{\text{max}} \exp\left[\frac{-1}{2 \sigma^2} \left( (x-x_{\text{peak}}(t_i))^2 + (y-y_{\text{peak}}(t_i))^2 \right)\right] \, dx \, dy + V_{kl}(t_i)
\]

where

\[I_{\text{max}} = \text{the maximum intensity received from target}\]

\[\sigma^2 = \text{the dispersion of the Gaussian intensity function}\]

This would be the measurement vector for a single Gaussian
hotspot target. For simulation of the multiple hotspot targets of interest for this research, three exponentials of Gaussian form are summed in place of the single Gaussian of Equation (3-30a).

**Spatially Correlated Background Noise**

The existence of spatial correlations in the background of each FLIR data frame has been documented by Harnly and Jensen (6:14). To discuss the generation of the spatially correlated noise, a numbering system for each pixel of an 8 x 8 input array is adopted (Figure 4). The 8 x 8 FLIR data array is numbered, as done by Harnly and Jensen (6:19) and Singletery (4:21) by pixels from 1 to 64 starting in the upper left-hand corner and proceeding across the rows.

In the 64 x 64 correlation coefficient matrix corresponding to a noise vector with each component associated with a single such pixel, each element relates the noise in that pixel with the noise in another pixel. For this research the spatial correlations are modelled as extending to the first and second nearest pixels. The element in row 1 and column 1 of the correlation coefficient matrix relates the noise in pixel one to itself and is therefore one. Similarly, the entire diagonal of the correlation coefficient matrix is equal to one. The correlation coefficient matrix is symmetric since the correlation $r_{ij}$ of the noise in pixel $i$ with noise of pixel $j$ is the same as that of the noise in
pixel \( j \) with the noise in pixel \( i \) \( (r_{ij} = r_{ji}) \).

The exponentially decaying and radially symmetric form of the correlation function was used in generating the coefficients of the first and second nearest neighbor correlation matrix, Equation (3-31), was used to generate the coefficient matrix:

\[
r_{ij} = \exp(-d_{ij})
\]

where \( d_{ij} \) = distance in # pixels from pixel \( i \) to pixel \( j \). Equation (3-31) implements a correlation distance of one pixel. The fact that nonzero values are computed only for the first and second neighbors results in only the coefficients relating a pixel's noise to its 24 nearest neighbors being nonzero (Figure 5).
In Figure 5, $r_{k,l}$ is the coefficient relating noise components in pixel $k$ to those of pixel $l$. Of the 25 nonzero values for any row or column, only six of them are distinct: one value of 1 for the correlation of a pixel with itself, four values of .3679 for pixels one pixel away, four values of .2431 for pixels $\sqrt{2}$ pixels away, four values of .1353 for pixels two pixels away, eight values of .1069 for pixels $\sqrt{5}$ pixels away, and four values of .0591 for pixels $\sqrt{8}$ pixels away. The matrix of correlation coefficients becomes

$$\mathbf{E} = \begin{bmatrix}
1 & r_{1,2} & r_{1,3} & \cdots & r_{1,64} \\
r_{1,2} & 1 & r_{2,3} & \cdots & r_{2,64} \\
r_{1,3} & r_{2,3} & 1 & \cdots & r_{3,64} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
r_{1,64} & r_{2,64} & r_{3,64} & \cdots & 1
\end{bmatrix}$$
Using the simplified noise covariance matrix of Harnly and Jensen, (6:18), the process for completing the generation of the noise covariance matrix is to premultiply the correlation coefficient matrix by the variance of the background noise.

\[ R = \sigma_b^2 \begin{bmatrix} 1 & r_{1,2} & r_{1,3} & \cdots & r_{1,64} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{1,64} & r_{2,64} & r_{3,64} & \cdots & 1 \end{bmatrix} \]

Once the correlation matrix, \( R \) is known, specific spatially correlated realizations of the 64 x 1 noise vector, \( \mathbf{v}(t_i) \) can be computed by using a Cholesky square root decomposition of \( R \) and a zero mean unit-variance independent Gaussian noise generator (see Appendix E). The noise vector realization will be added to the 8 x 8 input array. This noise vector is created by

\[ \mathbf{v}(t_i) = \sqrt{R} \mathbf{v}'(t_i) \quad (3-32) \]

where

\[ \mathbf{v}'(t_i) = 64 \text{ dimensional white Gaussian vector; each component zero mean, unit variance, and independent of other components (see Appendix E)} \]

\[ \sqrt{R} = \text{Cholesky square root} \]

The Cholesky square root decomposition (Chapter 7 reference 13) produces a lower triangular matrix such that

\[ \sqrt{R} \sqrt{R}^T = R \quad (3-33) \]
By using this formulation of the noise vector, Equation (3-32), the covariance of $v(t_i)$ is shown to be equal to the correlation matrix.

$$E[v(t_i)v^T(t_j)] = E[C_R v'(t_i)v'^T(t_j) C_R^T]$$

$$= C_R E[v'(t_i)v'^T(t_j)] C_R^T$$

$$= C_R I C_R^T \delta_{ij}$$

$$= R \delta_{ij} \quad (3-34)$$

Summary of Truth Model

This chapter presented the mathematical models used to account for target dynamics, atmospheric jitter effects, and the background and FLIR noises. These models described the translational position changes of the target's intensity distribution resulting from any of these reasons. The mathematical models were then summarized by transforming them into state space notation. Equation (3-25) showed how the truth model state space states are propagated forward in time while still accounting for the stochastic nature of the problem. Information on the computer creation of the measurement data for the computer simulation was also provided along with the model for spatial correlations of background noise.
IV. Extended Kalman Filter

Introduction

The previous chapter presented the truth model which is the best representation of the environment from which the measurement vector $z(t_i)$ is generated. An extended Kalman filter is used to process these measurement vectors to provide an estimate of the true target centroid location for the next sample time (for use by the controller) as well as of current state vectors needed for centering in the intensity image processing. A four-state extended Kalman filter model which accounts for time-corrupted dynamics and the bandwidth characteristics of the atmospheric jitter, as developed by Mercier, was used for this research. A wide variety of targets may be represented with this model by choosing appropriate noise strengths and correlation times.

Target Dynamics and Atmospheric Jitter Filter Models

The target dynamics and the atmospheric jitter are modelled as stationary, first order Gauss-Markov processes. These processes are generated as the outputs of first-order lags driven by white Gaussian noise (5:19). The differential equation which describes any of these modelled states is

$$\dot{x}_f(t) = \frac{1}{T_f} x_f(t) + w_f(t)$$

(4-1)
\[ E[w_f(t)] = 0 \quad (4-2) \]

\[ E[w_f(t)w_f(t+\tau)] = \frac{2\sigma_f^2}{T_f} \delta(\tau) \quad (4-3) \]

and

\[ T_f = \text{correlation time} \]

\[ w_f(t) = \text{white Gaussian noise process} \]

\[ \sigma_f = \text{root mean squared value of } x \]

The extended Kalman filter uses this stationary, first order, Gauss-Markov process to model the target dynamics and the atmospheric jitter in the x and y directions of the FLIR image plane. The stochastic differential equation model for each state upon which the filter is based is Equation (4-1). This is the same structure as Equations (3-2) and (3-3) except that a reduced-order model is used for atmospherics.

**State Space Model**

The four filter states will now be put into state space notation. The state vector differential equation which describes the four filter states \((x_d \ x_a \ y_d \ y_a)^T\), is

\[ \dot{X_f}(t) = F_f \ X_f(t) + W_f(t) \quad (4-4) \]

where

\[ F_f = \text{filter plant matrix} \]

\[ X_f(t) = \text{filter state vector} \]

\[ W_f(t) = \text{input white Gaussian noise vector} \]
Using the information from the previous section Equation (4-4) becomes

\[
\dot{x}_f(t) = \begin{bmatrix}
\frac{-1}{T_{df}} & 0 & 0 & 0 \\
0 & \frac{-1}{T_{af}} & 0 & 0 \\
0 & 0 & \frac{-1}{T_{df}} & 0 \\
0 & 0 & 0 & \frac{-1}{T_{af}} \\
\end{bmatrix} x_f(t) + w_f(t) \tag{4-5}
\]

where

\[T_{df} = \text{correlation time assumed for target dynamics}\]
\[T_{af} = \text{correlation time assumed for atmospheric jitter}\]

And

\[
E(w_f(t)) = \mathbf{0} \tag{4-6}
\]
\[
E(w_f(t)w_f^T(t+\tau)) = Q_f \delta(\tau) \tag{4-7}
\]

and

\[
Q_f = \begin{bmatrix}
\frac{2\sigma^2_{df}}{T_{df}} & 0 & 0 & 0 \\
0 & \frac{2\sigma^2_{af}}{T_{af}} & 0 & 0 \\
0 & 0 & \frac{2\sigma^2_{df}}{T_{df}} & 0 \\
0 & 0 & 0 & \frac{2\sigma^2_{af}}{T_{af}} \\
\end{bmatrix} \tag{4-8}
\]

\[\sigma^2_{df} = \text{assumed target dynamics noise variance}\]
\[\sigma^2_{af} = \text{assumed atmospheric jitter noise variance}\]
State Propagation

The extended Kalman filter must propagate its state estimate vector and conditional covariance matrix from one sample time to the next. The extended Kalman filter model state equations are linear in this application, and so the standard Kalman filter propagation equations can be used to propagate the filter states between sample times. These standard propagation equations are

\[ \hat{x}(t_{i+1}^-) = \Phi_f(t_{i+1}, t_i) \hat{x}(t_i) \]  \hspace{1cm} (4-9) \\
\[ P(t_{i+1}^-) = \Phi_f(t_{i+1}, t_i) P(t_i^+) \Phi_f^T(t_{i+1}, t_i) + \int_{t_i}^{t_{i+1}} \Phi_f(t_{i+1}, \tau) Q_f \Phi_f^T(t_{i+1}, \tau) d\tau \] \hspace{1cm} (4-10)

where

\[ \Phi_f(t_{i+1}, t_i) = \text{filter state transition matrix} \]

\[ P(t_i^+) = \text{conditional state covariance matrix from measurement update equation at time } t_i \]

\[ P(t_{i+1}^-) = \text{conditional state covariance matrix propagated from time } t_i \text{ to } t_{i+1} \]

\[ Q_f = \text{noise covariance kernel descriptor given in Equation (4-8)} \]

The filter's state transition matrix \( \Phi_f(t_{i+1}, t_i) \) must satisfy the differential equation

\[ \dot{\Phi}_f(t, t_i) = F_f \Phi_f(t, t_i) \] \hspace{1cm} (4-11)

over the interval \((t_i, t_{i+1})\), subject to the initial condition
\[ \Phi_f(t_{i+1}, t_i) = I \] (identify matrix) \hspace{1cm} (4-12)

The time-invariant plant matrix, \( P_f \), and fixed sample period result in a constant state transition matrix, \( \Phi_f(t_{i+1}, t_i) \), for any given sample period:

\[ \Phi_f(t_{i+1}, t_i) = e^{P(t_{i+1} - t_i)} = e^{P \Delta t} \]

\[
\begin{bmatrix}
    e^{-\Delta t/T_{df}} & 0 & 0 & 0 \\
    0 & e^{-\Delta t/T_{af}} & 0 & 0 \\
    0 & 0 & e^{-\Delta t/T_{df}} & 0 \\
    0 & 0 & 0 & e^{-\Delta t/T_{af}}
\end{bmatrix} \hspace{1cm} (4-13)
\]

where \( \Delta t \) is the sample period, \( \Delta t = t_{i+1} - t_i \).

The solution to the integral term of Equation (4-10) becomes

\[
Q_D = \begin{bmatrix}
    \sigma_{df}^2 (1 - e^{-2 \Delta t/T_{df}}) & 0 & 0 & 0 \\
    0 & \sigma_{af}^2 (1 - e^{-2 \Delta t/T_{af}}) & 0 & 0 \\
    0 & 0 & \sigma_{df}^2 (1 - e^{-2 \Delta t/T_{df}}) & 0 \\
    0 & 0 & 0 & \sigma_{af}^2 (1 - e^{-2 \Delta t/T_{af}})
\end{bmatrix} \hspace{1cm} (4-14)
\]

The matrix of Equation (4-14) is constant for a given sampling time \( \Delta t \), due to system time invariance and noise stationarity. The values of \( \sigma_{df}^2 \) and \( \sigma_{af}^2 \) of the \( Q_f \) matrix of Equation (4-8)
are determined during an off-line tuning process. This off-line tuning produces the values which optimize tracking performance.

In summary, the estimated state vector and the conditional covariance matrix propagation equations are

\[
\hat{X}(t_{i+1}^-) = \begin{bmatrix}
    e^{-\Delta t/T_{df}} & 0 & 0 & 0 \\
    0 & e^{-\Delta t/T_{af}} & 0 & 0 \\
    0 & 0 & e^{-\Delta t/T_{df}} & 0 \\
    0 & 0 & 0 & e^{-\Delta t/T_{af}}
\end{bmatrix} \hat{X}(t_i^+)
\]

(4-15)

\[
\hat{P}(t_{i+1}^-) = \begin{bmatrix}
    e^{-\Delta t/T_{df}} & 0 & 0 & 0 \\
    0 & e^{-\Delta t/T_{af}} & 0 & 0 \\
    0 & 0 & e^{-\Delta t/T_{df}} & 0 \\
    0 & 0 & 0 & e^{-\Delta t/T_{af}}
\end{bmatrix} \hat{P}(t_i^+)
\]
Equations (4-15) and (4-16) propagate the four filter states between sample times. Once the predicted states for the next sample time are calculated, the estimates of the target's position \( x_d(t_{i+1}) \) and \( y_d(t_{i+1}) \), are fed to a feedback controller. The controller positions the FLIR to be centered on this estimated target location by that next sample time \( t_{i+1} \). In view of Equation (3-16), the filter expects the apparent target's intensity function to be offset from the center of that field-of-view by the estimated jitter effects also calculated by Equation (4-15).

**Measurement Update Equations**

Equation (3-30) can be rewritten in the general form

\[
z_k(t_i) = h_{kl}(x(t_i), t_i) + v_{kl}(t_i)
\]

(4-17)

where \( h_{kl}(x(t_i), t_i) \) is a nonlinear function of the states at sample time \( t_i \). Equation (4-17) represents each pixel's information as a nonlinear function of the states plus additive corrupting noise, \( v_{kl}(t_i) \). Once the Kalman filter's
state vector and covariance matrix are propagated to the next sample time, the filter uses this propagated state estimate along with the information contained within each of the 64 pixels represented by Equation (4-17), to compute an updated estimate of the underlying target centroid location. To process the information contained within the measurement vector, the extended Kalman filter was used, since it can handle the nonlinearities of the measurement equation, Equation (4-17). Moreover, it is less computationally burdensome than other nonlinear filters, and computation time is of extreme importance in this application.

In the process of generating the appropriate filter gain, an extended Kalman filter linearizes the measurement equation about the most recent estimate of the states, \( \hat{x}(t^-) \). The inverse covariance form (13:257) of the extended Kalman filter measurement update equations is used to eliminate the need for the inversion of a 64 x 64 matrix for every update (5:26). This 64 x 64 inversion would be required during the calculation of the Kalman filter gain in the usual form of the update Equations (13:233), i.e.,

\[
K = P^{-1} H^T (H P^{-1} H^T + R)^{-1},
\]

because of the 64 scalar measurements. The extended Kalman filter update equations in inverse covariance form are

\[
P^{-1}(t_1^+) = P^{-1}(t_1^-) + H^T(t_1) R^{-1}(t_1) H(t_1) 
\]

\[
P(t_1^+) = [P^{-1}(t_1^+)]^{-1}
\]
\[
K(t_1) = P(t_1^+) H^T(t_1) R^{-1}(t_1)
\] (4-20)

\[
\hat{x}(t_1^+) = \hat{x}(t_1^-) + K(t_1)[z(t_1) - h(\hat{x}(t_1^-), t_i)]
\] (4-21)

where

\[
H(t_1) = \frac{\partial h(\hat{x}(t_1^-), t_i)}{\partial \hat{x}} = \text{linearized function of intensity measurements (see Appendix F)}
\] (4-22)

\[
P(t_1^-) = \text{propagated conditional covariance matrix before measurement incorporation at time } t_i
\]

\[
\hat{x}(t_1^-) = \text{propagated state estimate vector of filter before measurement incorporation at time } t_i
\]

\[
K(t_1) = \text{Kalman filter gain}
\]

\[
h(\hat{x}(t_1^-), t_i) = \text{nonlinear function of intensity measurements at time } t_i, \text{ as function of filter state estimates } \hat{x}(t_1^-)
\]

\[
z(t_i) = \text{actual realization of measurement vector}
\]

This formulation of the extended Kalman filter update equations requires only two 4 x 4 inverses, of the \(P(t_1^-)\) matrix and the \(P^{-1}(t_1^+)\), and therefore eases the computational burden of the algorithm substantially. The inversion of the \(R(t_i)\) matrix is accomplished only once, potentially offline, and the result of that inversion, \(R^{-1}(t_i)\), is stored for use in Equation (4-18). Moreover, in the spatially uncorrelated pixel noise case, \(R^{-1}(t_i)\) is of the form \((1/R)I\).

In this research, the nonlinear \(h\) function and its derivatives with respect to the filter states are derived using the FFT, phase shifting and smoothing techniques dis-
cussed in Chapter II. The structure of this algorithm is shown in Figure 1.

In summary, the inverse covariance form of the extended Kalman filter measurement update equations were used in this research to ease the computational burden. Equations (4-18) through (4-21) present the equations implemented in the computer simulation.

Summary of the Extended Kalman Filter Equations

This section summarizes the propagation and update equations used for this research

**Propagation Equations**

\[
\hat{x}(t_i^-) = \phi_f \hat{x}(t_i^-) + \hat{P}(t_i^-) \phi_f^T + Q_f
\]

where

\[
\phi_f = \begin{bmatrix}
  e^{-\Delta t/T_d} & 0 & 0 & 0 \\
  0 & e^{-\Delta t/T_a} & 0 & 0 \\
  0 & 0 & e^{-\Delta t/T_d} & 0 \\
  0 & 0 & 0 & e^{-\Delta t/T_a}
\end{bmatrix}
\]

and where
\[
\mathbf{Q}_D = \begin{bmatrix}
\sigma^2_{df}(1-e^{-2\Delta t/T_{df}}) & 0 & 0 & 0 \\
0 & \sigma^2_{df}(1-e^{-2\Delta t/T_{af}}) & 0 & 0 \\
0 & 0 & \sigma^2_{df}(1-e^{-2\Delta t/T_{df}}) & 0 \\
0 & 0 & 0 & \sigma^2_{df}(1-e^{-2\Delta t/T_{af}})
\end{bmatrix}
\] (4-26)

**Estimate of states as modified by controller action:**

\[
\hat{\mathbf{x}}'(t_i^-) = \begin{bmatrix}
0 \\
\mathbf{x}_a(t_i^-) \\
0 \\
\hat{y}_a(t_i^-)
\end{bmatrix}
\] (4-27)

**Update Equations**

\[
P^{-1}(t_i^+) = P^{-1}(t_i^-) + H^T(t_i) \ R^{-1}(t_i) \ H(t_i)
\] (4-28)

\[
P(t_i^+) = [P^{-1}(t_i^+)]^{-1}
\] (4-29)

\[
K(t_i) = P(t_i^+) \ H^T(t_i) \ R^{-1}(t_i)
\] (4-30)

\[
\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}'(t_i^-) + K(t_i) \begin{bmatrix}
\mathbf{z}(t_i^-) - h(\hat{\mathbf{x}}'(t_i^-), t_i)
\end{bmatrix}
\] (4-31)
V. Correlator-Kalman Filter Tracker

Introduction

The Kalman filter operating upon the raw digitized image has been shown to perform well in comparison to standard correlation trackers (5;6). In these studies, the filter had valid apriori information about the analytic form of the intensity function, \( h(\hat{x}(t_i), t_i) \). The performance of the Kalman filter operating upon the raw digitized data versus a correlator should be less beneficial to the filter without such apriori intensity function information. It is also computationally easier to implement a correlation algorithm than to implement a high measurement dimension extended Kalman filter.

This chapter presents an alternate tracking concept which is a hybrid in that it uses both correlation and Kalman filtering. A correlator is first used to generate position estimates for the target within a noise-corrupted frame of data. The correlator uses the estimated intensity function, \( h(\hat{x}(t_i), t_i) \), generated via the data processing algorithm of Chapter II, as its template. This is different from current algorithms which just use previous raw data as a template. This modification to the standard correlation algorithm is expected to enhance performance. The standard correlation algorithm does not exploit any knowledge of the target's dynamics or any knowledge of those disturbances which could cause apparent translational intensity function offsets.
The improved correlation algorithm developed for this research positions the template with the benefit of \( \hat{X}(t_{i+1}) \) which does exploit such knowledge. To account for correlation errors, the position estimates of the correlator are processed by a linear Kalman filter to produce better position estimates.

Section one of this chapter provides information on how the correlation algorithm was implemented. The next section statistically characterizes the errors in the position estimates of this implementation of the correlation algorithm. The last section of this chapter develops the linear Kalman filter which processes the correlator’s position estimates.

**Correlator Implementation**

This section presents the implementation of the correlation algorithm used to provide position "measurements" to the Kalman filter. The correlation routine, written for this research, computes the cross correlation of a template, the result of the data processing of Chapter II, and the noise-corrupted FLIR data frame. The FFT is used to compute the cross correlation (8:557) as shown below.

\[
F[g(x,y)] = G(f_x, f_y) 
\]

\[
F[\hat{G}(x,y)] = L(f_x, f_y) 
\]  

(5-1)  

(5-2)

\[
F[g(x,y) \ast \hat{G}(x,y)] = G(f_x, f_y) \ast L(f_x, f_y) 
\]  

(5-3)
where
\[ g(x,y) \ast \hat{f}(x,y) = \text{cross correlation of the two-dimensional spatial sequences } g(x,y) \text{ and } \hat{f}(x,y) \]
\[ \hat{f}^*(f_x,f_y) = \text{complex conjugate of the Fourier transform of the sequence } \hat{f}(x,y) \]

Taking the inverse transform of Equation (5-3) yields the cross correlation of the two-dimensional sequences \( g(x,y) \) and \( \hat{f}(x,y) \):

\[ R(x,y) = g(x,y) \ast \hat{f}(x,y) = F^{-1}[G(f_x,f_y) \cdot \hat{f}^*(f_x,f_y)] \]  (5-4)

where \( R(x,y) \) is the result of the correlation.

To show how the position estimates are obtained from this algorithm, a simple two-dimensional Gaussian array is examined. Let \( \hat{f}(x,y) \) represent the two-dimensional array which contains the template, which for this example is a perfect target replica and is a centered Gaussian function with \( \sigma^2 = 2 \) pixels within the two-dimensional array. Figure 6 is an example of a 24 x 24 pixel array with the template's intensity being represented by a gray-scale (see Subroutine Display Appendix H). Each dash around the perimeter of the field-of-view represents one pixel in Figure 6. The use of the 24 x 24 pixel array allows the 8 x 8 pixel tracking window to be padded by 8 rows and 8 columns of zeros or data for the computation of FFTs.

Similarly, let \( g(x,y) \) represent the two-dimensional array of noise-corrupted data, which contains the target's
offset intensity function. Figure 7 is an example of a 24 x 24 pixel array which is slightly noise corrupted. The target's single Gaussian intensity pattern is offset by one pixel in the horizontal direction from the center of the 24 x 24 pixel array. A gray scale representation of the result of the cross correlation, $R(x,y)$, is shown in Figure 8.

The expected result of correlating a Gaussian with a Gaussian is a Gaussian whose location within the resulting matrix, $R(x,y)$, indicates the relative pixel position offset between the template and the target. The two-dimensional array $R(x,y)$ is symmetric in the vertical direction which indicates no offset there. In the horizontal direction a one pixel circular shift to the left would restore symmetry. That is, if all of the columns of Figure 8 were shifted one column to the left and the leftmost column shifted into the rightmost column, horizontal symmetry would be restored. Obviously the information that the target is one pixel horizontally offset from the center of the two-dimensional noise-corrupted data array is contained in the result of the cross correlation, $R(x,y)$. The fact that the two-dimensional Fourier transform assumes that this two-dimensional sequence is one period in each direction of an infinitely periodic sequence explains the use of the circular shift. Inherent in the use of the FFT to derive $R(x,y)$ is the assumption that the next pixel to the left of any pixel in column one
Figure 6. Centered Single Gaussian Template
Figure 7. Noise Corrupted Data Array
| 10000X+ | X01 |
| 1000X+ | +X01 |
| 1000X+ | +X1 |
| 1XXX+  | +1  |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | 1   |
| 1      | ++  |
| 1XX++  | +1  |
| 1000X+ | +X1 |
| 10000X | +01 |

Figure 8. Result of Cross Correlation of Data and Template
is equal to the corresponding pixel in column 24. Using this fact to change the representation of the result of the cross correlation, the magnitudes of the pixels intensities of quadrant one are switched with those of quadrant four. Similarly, quadrant two's pixels are swapped with quadrant three's. The result of the quadrant swapping as a representation of the correlation output matrix, \( R(x,y) \), where the relative position offset between the template and the target is now represented by an offset of the resulting Gaussian's maximum from the center of the sequence. Figure 9 provides a gray scale representation of the result of the cross correlation, \( R(x,y) \), once the quadrants are swapped. Figure 8 corresponds to the convention of defining the origin for the correlation as corresponding to the origins of the original two figures. This means that the two figures' origins are initially superimposed to create the first correlation data point. The correlation information corresponding to the four pixels which surround the center of the fields-of-view of the original figures is contained in the four corner pixels in the correlation matrix. The convention on the definition of origins attained by "swapping" is where the center of the 24 x 24 correlation matrix corresponds to the correlations of the centers of the two original figures.

By comparing the original Gaussian template and the noise-corrupted data with the output of the cross correlation
Figure 9. Diagonal Quadrant Swap of Cross Correlation
given in Figure 9, a spreading of the function is observed. The noise has also induced some asymmetries in the sequence \( R(x,y) \). The magnitudes of the components of the sequence \( R(x,y) \), are a measure of the degree of resemblance between the template and the data sequences. If the magnitude of an element of the sequence \( R(x,y) \) is less than some preset fraction of the maximum value of any element in that sequence, then that element is considered to contain poor correlation information and can be set equal to zero. The result is that such elements will have no effect on the computed offset between the template and the target. Thresholding is often done to suppress lower peaks in the result of the correlations. Figure 10 shows the result of setting this fraction to .5 for the sequence of Figure 9.

Once the threshold has smoothed the result of the cross correlation, a peak detector is usually used to find where the maximum resemblance of the two sequences occurs. Instead of a peak detector, a centroid summation was used to find the center of mass of the two-dimensional cross correlation sequence, \( R(x,y) \). The center of mass of the thresholded correlation is assumed to be good indication of the peak location. In either direction, horizontal or vertical, the centroid summation is defined by
Figure 10. Result of Thresholding of .5
where \( i \) is the horizontal or vertical coordinate of a given pixel, and \( \text{Amp}_i \) is the amplitude value for that pixel, and \( N \) is the total number of pixels in the array. For the horizontal direction, the horizontal coordinate's of all the elements of \( R(x,y) \) would be multiplied by the amplitudes of the respective elements and the products would be summed. The resulting summation, the numerator of Equation (5-5), would then be divided by the sum of all the amplitudes. Equation (5-5) is implemented in both directions providing a position estimate of the offset of the target from the center of the data frame.

**Correlator Error Statistics**

This section statistically characterizes the errors in the position estimates of the correlation algorithm presented in the previous section. These position estimates will be the measurements provided to a Kalman filter which will generate a better estimate of the target's position. The two-dimensional discrete-time measurement vector, \( z_c(t_i) \), is a linear combination of the variables of interest, the target intensity function's true position, but corrupted by an uncertain measurement disturbance \( v_c(t_i) \) also of dimension two.
\[ z_c(t_i) = H_c x_c(t_i) + v_c(t_i) \]  \hspace{1cm} (5-6)

where

\[ z_c(t_i) = \begin{bmatrix} x_{dc} \\ y_{dc} \end{bmatrix} = \text{the estimated x and y coordinates by} \\
\text{the correlation/centroid algorithm} \]

\[ H_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \text{the linear combination of state variables which contribute to the respective measurement elements} \]

\[ x_c = \begin{bmatrix} x_d \\ x_a \\ y_d \\ y_a \end{bmatrix} = \text{the state vector} \]

\[ v_c(t_i) = \text{additive noise corruption, assumed to} \\
\text{be white, and Gaussian, with statistics to be determined} \]

The additive noise corruption vector, \( v_c(t_i) \) of the measurement model of Equation (5-6), must account for the statistical effects of the errors in the correlator/centroid position estimates.

Software was developed to test the correlator/centroid algorithm's position estimates repeatedly, and histograms of the resulting error were generated. Figure 11 contains examples of the error histograms produced for a target offset by .3 pixels from the template in the horizontal direction only. The errors could be modelled as Gaussian random variables with appropriate means and variances. The values needed to describe the error's mean or variance would be a function of the template-target separation, the threshold level used by the centroid and even the variance of the
Figure 11a. Histogram of Error in Horizontal Position Estimate
background noise. For this research a value for the threshold of .5 and a target-template separation of .15 pixels were chosen to establish error statistics. Other values which were considered for the threshold were .1, .2, .25, .75, and even .90. The value of .5 was chosen since it produced acceptable results for the signal-to-noise ratios of interest. Error data for other choices of thresholds will be provided in the next chapter. At the sampling rate of 30Hz, for the trajectories being simulated in this research, a propagation error of not more than .15 pixels was expected. Background noise was allowed to vary over the full range of signal-to-noise ratios of 10 to 20. The result was a mean error of approximately -.080 pixels and a variance of .00363 pixels$^2$ in the horizontal direction. The error in the vertical direction has a mean of .116 pixel with a variance of .00598 pixels$^2$. These values can now be used to describe statistically the errors in the correlator/centroid position estimates. This characterization is needed for the measurement model of Equation (5-6).

The difference in the vertical and horizontal errors above is because of the locations of the three Gaussians of the target intensity profile. If the center of the 8 x 8 pixel tracking window is located at coordinates (0,0) then the Gaussians would be located at (0,-2.667), (-2,1.333), and (+2,1.333). This placement forces the center of mass of
the intensity profile to coordinates (0,0), but the resulting intensity pattern is not radially symmetric about (0,0).

**Kalman Filter Tracker**

The mathematical models for target dynamics and atmospheric jitter which were developed in Chapter IV are also used in the Kalman filter which processes the correlator's intensity function position estimates. These models accounted for time correlated dynamics and the bandwidth characteristics of the atmospheric jitter. The standard Kalman filter propagation equations, as given by Equations (4-23) through (4-26), are also used by this Kalman filter. The difference in the Kalman filter used in this alternative tracker is that it processes a two-dimensional measurement vector of intensity function position estimates, whereas the previous filter processed 64 intensity measurements. The measurement equation for the new filter was given in Equation (5-6). The low dimensionality of the measurement vector eliminates the need and in fact the applicability of the inverse covariance form of the Kalman filter update equations. The correlator/centroid intensity function's position estimates, which constitute the two-dimensional measurement vector, are linear combinations of the Kalman filter states, as seen in Equation (5-6). The linear filter measurement update equations used are given in Equations (5-7) through (5-9).
\[ K(t_i) = P(t_i^-)H^T(t_i) [H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]^{-1} \]  
(5-7)

\[ \hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i) [z(t_i) - H(t_i)\hat{x}(t_i^-)] \]  
(5-8)

\[ P(t_i^+) = P(t_i^-) - K(t_i)H(t_i)P(t_i^-) \]  
(5-9)

The values of \( P(t_i^-) \) and \( \hat{x}(t_i^-) \) are obtained from the propagation equations and the measurement uncertainty covariance matrix, \( R(t_i) \), is assumed diagonal with the values determined as explained in the previous section. The diagonal nature of \( R(t_i) \) assumes that the correlation position estimate uncertainty in one direction is independent of the uncertainty in the position estimate in the other direction. Since the errors are a function of SNR and threshold level, this independence assumption is subject to later revision. Direct computation of the off-diagonal elements of \( R(t_i) \) could verify or invalidate the assumption of independence of the errors in the position estimates.

In summary, this chapter developed an alternate tracking algorithm which uses a correlation algorithm to provide intensity function position estimates as a measurement vector to a Kalman filter. This algorithm is different from standard correlation trackers in that it uses the model of target dynamics from the Kalman filter to position the template, thresholding to remove false peaks, on-line derived template shape, and correlator position estimate enhancement via a Kalman filter. The Kalman filter uses internal models to separate what
translational motion of the intensity function resulted from target dynamics and what resulted from atmospheric jitter. The results from this tracking algorithm and the algorithm of Figure 1 are given in the next chapter.
VI. Performance Analysis

Introduction

This chapter presents the results from the testing of the tracking algorithms developed in the previous chapters. The first section of this chapter derives figures of merit which provide a means to evaluate the accuracy in the derivation of the intensity functions contained in Chapter II. This accuracy is ultimately only important in its impact on tracking ability. The sensitivity of tracking performance to the accuracy of the derived intensity functions will be shown as well. The tracking ability performance criteria are presented in the second section of this chapter. The third section discusses those variable parameters which the computer simulation allows the input to change for sensitivity analysis. The errors in the derivation of the intensity functions and the tracking errors are plotted as mean ± one sigma (standard deviation) errors in the next section. The final section of this chapter condenses the results of this research into tables. These tables cross reference variations in the parameters which control the Kalman filter and the pattern recognition process to tracking and intensity function derivation errors. Important trends and sensitivities are revealed in these tables. The statistical information of this chapter was generated using Monte Carlo techniques (13:329), see Appendix G.
Derivation of Intensity Functions

The accuracy of the on-line derivation of the intensity functions (see Chapter II) is actually of importance here only since it affects tracking ability. To make some reasonable judgement on how good a representation of \( h(\hat{x}(t_i), t_i) \) and \( H(\hat{x}(t_i), t_i) \) has been derived, a simple figure of merit is developed in this section.

At each frame the truth model provides information on the location of the centroid of the true target's intensity functions. The data processing of Chapter II derives on-line estimates of these intensity functions. At each frame the difference between the true target intensity functions:

1. the nonlinear intensity function, \( h_t \)
2. the derivative of \( h_t \) with respect to a change in either horizontal state, \( x_d \) or \( x_a \), \( H_xt \)
3. the derivative of \( h_t \) with respect to a change in either vertical state, \( y_d \) or \( y_a \), \( H_yt \)

and the estimated intensity functions, \( \hat{h}_t \), \( \hat{H}_x \), and \( \hat{H}_y \), respectively, are computed pixel by pixel.

As the simulation progresses, at each frame \( i \) two 8 x 8 arrays are maintained to compute the accuracy of \( \hat{h}_t \). One array is the difference between \( h_{ti} \) and \( \hat{h}_i \). The square of this error is also maintained: \( (h_{ti} - \hat{h}_i)^2 \) where \( i \) represents which frame is being processed.

Many time histories are processed in the Monte Carlo study (as described in detail in Appendix G), and for each...
frame the mean and variance of the error associated with each pixel for that frame is desired. The mean error for a given pixel, \( j = 1, \ldots, 64 \), for a given time frame, \( i = 1, \ldots, 20 \), over all the time histories run, \( k = 1, \ldots, N \), would be given by

\[
M_{e ij} = \frac{1}{N} \sum_{k=1}^{N} (h_{t ijk} - \hat{h}_{ijk}) = \frac{1}{N} \sum_{k=1}^{N} e_{ijk} \tag{6-1}
\]

where

\[
M_{e ij} = \text{mean error for pixel } j \text{ at the time frame } i, \quad \text{averaged over the } N \text{ simulations.}
\]

\( N \) = number of simulations

\( k \) = index of Monte Carlo simulation runs, going from 1 to \( N \)

\( h_{t ijk} \) = value for frame \( i \), pixel \( j \), of the truth model

\( \hat{h}_{ijk} \) = intensity for the \( k \)th simulation

\( e_{ijk} \) = estimate value for frame \( i \), pixel \( j \), of the intensity as derived in the \( k \)th simulation

The variance of the error for frame \( i \), pixel \( j \), would be given by

\[
\sigma_{e ij}^2 = \frac{1}{N} \sum_{k=1}^{N} (e_{ijk} - M_{e ij})^2
\]

\[
= \frac{1}{N} \sum_{k=1}^{N} (e_{ijk}^2 - 2M_{e ij}e_{ijk} + M_{e ij}^2)
\]

\[
= \frac{1}{N} \sum_{k=1}^{N} e_{ijk}^2 - 2M_{e ij} \cdot \frac{1}{N} \sum_{k=1}^{N} e_{ijk} + \frac{1}{N} \sum_{k=1}^{N} M_{e ij}^2
\]

\[
= \frac{1}{N} \sum_{k=1}^{N} e_{ijk}^2 - 2M_{e ij} \cdot M_{e ij} + \frac{1}{N} \cdot N \cdot M_{e ij}^2
\]

\[
= \frac{1}{N} \sum_{k=1}^{N} e_{ijk}^2 - M_{e ij}^2 \tag{6-2}
\]

73
Note that in general \( l/(N-1) \) could be used in place of \( 1/N \) in the first line of this equality to reduce the bias in this estimate, but with \( N=20 \) this substitution would not make much difference. Equations (6-1) and (6-2) show that two \( 8 \times 8 \) arrays must be maintained for each frame for each intensity function of interest. In this research, the errors in \( \hat{h}, \hat{h}_x, \) and \( \hat{h}_y \) are of interest, and so six arrays were maintained for each frame.

For this research, new data frames are generated at a 30 Hz rate. If, as in Appendix G, twenty frames constitute a single tracking time history, two-thirds of a second simulation time, then six \( 8 \times 8 \times 20 \) arrays are needed. At the end of the simulation these six \( 8 \times 8 \times 20 \) arrays contain the information needed to compute the ensemble average and variance of errors for any pixel, \( j \), for any frame, \( i \). With the assumption that every pixel is as important as every other pixel, the spatial average of the mean pixel errors can be computed. Equation (6-3) shows how to compute the spatial average of the mean pixel errors for any frame, \( i \).

\[
M_{e_i} = \frac{1}{64} \sum_{j=1}^{64} M_{e_{ij}} \quad (6-3)
\]

where

\[
M_{e_i} = \text{the spatial average (over all the pixels) of the mean pixel errors at the } i\text{th frame}
\]

\[
M_{e_{ij}} = \text{mean (i.e. ensemble average over all the simulations) at the } j\text{th pixel for the } i\text{th frame}
\]
Note that if each pixel were not considered equally important, a weighted average could replace (6-3). Similarly, the spatial average of the variance for any frame \( i \) could be computed by

\[
\sigma_{e_i}^2 = \frac{1}{64} \sum_{j=1}^{64} \sigma_{e_{ij}}^2
\]

(6-4)

where

\[
\sigma_{e_i}^2 = \text{the spatial average of all the ensemble pixel error variances, over all the pixels at the } \text{ith frame}
\]

\[
\sigma_{e_{ij}}^2 = \text{variance of the error of the } j\text{th pixel at the } \text{ith frame}
\]

The benchmark for intensity function derivation errors is found by providing perfect information of \( \hat{x}_{\text{peak}} \) and \( \hat{y}_{\text{peak}} \) to the pattern recognition algorithm. This benchmark is displayed as the twelfth entry of Table I in the next chapter.

**Tracking Ability**

Appendix G describes the Monte Carlo study used to generate the statistics of this section. The quantities of interest, with respect to tracking, are the errors in the estimated value of \( \hat{x}_d(t_i^-), \hat{y}_d(t_i^-), \hat{x}_d(t_i^+), \text{ and } \hat{y}_d(t_i^+) \). It is also important to estimate the location of the intensity distribution's centroid. The accuracy of the estimation of the centroid's location, \( \hat{x}_{\text{peak}}(t_{i+1}^-) \) and \( \hat{y}_{\text{peak}}(t_{i+1}^-) \) of Equation (3-30a), will affect the accuracy of the estimated intensity functions and in-turn affect the tracking accuracy.
By comparing the errors before and after incorporation of a measurement, information about how the estimates were improved from the information of the data frame is obtained.

Figure G-1 shows how the error samples are generated. Similar to what was done in the previous section, error statistics are calculated as the mean error for the variable of interest at frame i,

\[ \overline{E}_{x_{d_i}} = \frac{1}{N} \sum_{k=1}^{N} (x_{t_{d_{ik}}} - \hat{x}_{d_{ik}}) = \frac{1}{N} \sum_{k=1}^{N} x_{d_{ik}} \]  

(6-5)

where

\[ \overline{E}_{x_{d_i}} = \text{mean error (i.e., ensemble average error over all simulations) in } x \text{ dynamics for frame } i \]

\[ x_{t_{d_{ik}}} = \text{truth model, } x\text{-dynamics value at frame } i \text{ for simulation } k \]

\[ \hat{x}_{d_{ik}} = \text{filter estimated } x\text{-dynamic value at frame } i \text{ for simulation } k \]

and the variance of the errors,

\[ \sigma^2_{e_{x_{d_i}}} = \frac{1}{N} \sum_{k=1}^{N} e_{x_{d_{ik}}}^2 - \overline{E}^2_{x_{d_i}} \]  

(6-6)

The \(1/(N-1)\) substitution discussed in the previous section is also applicable here. The generalization of equations (6-5) and (6-6) to compute the errors in \(\hat{y}_d\), \(\hat{x}_{\text{peak}}\), or \(\hat{y}_{\text{peak}}\) is clear. The benchmark for tracking performance is set by providing perfect \(h\), \(H_y\), and \(H_x\). This is the case studied by Capt Mercier (5). Table II of that research (5:57) lists a mean error of .2 pixels for a SNR = 20. That value can be
ENHANCED TRACKING OF AIRBORNE TARGETS USING FORWARD LOOKING INF—ETC(U)

S K ROGERS
AFIT/CEO/EE/81D-5
compared to entry one of Table I, contained in the next section, to show only a slight degradation in tracking ability when deriving the intensity profiles.

Variation of Parameters

The computer simulation was developed to allow certain filter and data processing parameters to be varied. This was accomplished in order to test the tracking algorithms of Figure 1 and Chapter V with different design characteristics and in different tracking environments.

The first parameter to be varied is the spread parameter, $\sigma_g^2$, of the target's Gaussian intensity profiles. This value is used by the truth model to generate the three-Gaussian hot spot data each having the given spread, (see Equation (3-35)). The standard value for this parameter, for this research, was 2.0 but runs were also made with values of 1.0 and 3.0.

The next parameter is the number of zeros to pad around the data. Padding of the finite data array with zeros is a common engineering practice to allow manipulation of the transformed data array to provide results compatible with the limited field-of-view. In Chapter II it was shown that DFTs assume that the finite data array is one period of an infinitely periodic two-dimensional sequence. Any manipulation of the transformed data array uses this assumption of infinite periodicity. Padding with 8 rows and 8 columns of
zeros creates a 24 x 24 pixel array which is assumed to be one period by the DFT. The padding insures that the infinite periodicity assumption will not affect results within the 8 x 8 tracking window. If the spread parameter, \( \sigma_g^2 \), is chosen such that the target intensity height is approaching zero near the edge of the 8 x 8 tracking window, then padding with zeros will not adversely affect the results of the data processing of Figure 1. However, if \( \sigma_g^2 \) is such that significant intensity magnitudes exist outside the 8 x 8 array, to pad with zeros arbitrarily would induce an artificial edge in the intensity function. These edges in the two-dimensional spatial intensity array will cause increased magnitudes of the high frequency components in the transform domain. For this application, a full frame of FLIR data actually consists of 300 x 400 pixels, so, when necessary, the 8 x 8 array could be padded with the noise-corrupted data instead of zeros. When this parameter is set to eight, then the 8 x 8 tracking window is surrounded by 8 rows and 8 columns of zeros to fill up a 24 x 24 complex data array. The standard value, for this research, for this parameter is zero causing the 8 x 8 tracking window to be padded by noise-corrupted data.

The next two parameters are the number of frames per simulation and the number of simulations per Monte Carlo study. Appendix G explains these parameters in terms of the Monte Carlo study. Both of these parameters were set to 20.
Early in the testing of the tracking algorithms, fifty frames were used to insure steady-state error was reached in a 20 frame time history. The consistency of the computed statistics, between twenty and fifty frame time histories motivated the use of the twenty frame simulation.

Alpha, the relative weighting parameter for the smoothing process, is the next parameter which may be varied. Chapter II explains how alpha affects the smoothing process, and Equation (2-9) explicitly displays its role in the algorithm. For this research, the standard value for alpha is .1, but a value for alpha of .1 and .2 are also investigated. Using alpha equal to 1. results in no smoothing and thus provides a benchmark to demonstrate what benefit smoothing provides.

The number of high frequency components of the FFT of the image to zero out is the next parameter. This provides the ability to investigate how spatial filtering within the intensity function derivation portion of Figure 1 could enhance or corrupt tracking performance. Spatial filtering could easily be accomplished within the optical implementation discussed in the next chapter if it is shown beneficial here. For standard runs, no frequencies were zeroed. Runs zeroing the two and four highest spatial frequency components were accomplished also.

The input background noise variance, $\sigma_b^2$, is next.
This value also includes FLIR noise contributions. Since the maximum values of the three Gaussian intensities were twenty, the signal to noise ratio, SNR, is equal to

\[
\text{SNR} = \frac{\text{peak signal value}}{\text{rms background noise}} = \frac{20}{\sigma_b^2}
\]

SNR values of 20 (standard) to 10 were considered as representative of realistic tracking scenarios.

The next two parameters are filter parameters which are varied to tune the filter for optimal tracking. The variance of the dynamic discrete time noise driving the target states of the filter (see Equation 4-8) is a measure of the uncertainty of the filters dynamics model. The variance of the atmospherics for the filter (see Equation 4-8) is similarly explained. Without any in-depth tuning to optimize tracking performance, values of .1 and .001 pixels\(^2\) were used respectively. The disparity in these values indicates the filter's atmospheric model is expected to be a very good representation of the atmospheric disturbance. The approximation of ignoring the high frequency pole made by the Kalman filter is not expected to affect the filter's atmospheric representation to any substantial degree based on performance analyses of previous filters that incorporated this same approximation (4;5;6). These values did establish acceptable tuning. Chapter VIII, Recommendations, discusses issues of tuning which should eventually be considered.
The next parameter is the RMS atmospheric jitter induced in pixels (see Equation 3-8). A value of .1 pixel jitter was used as a standard.

The software also allows the option of which domain the exponential smoothing algorithm is to be accomplished. Appendix C discusses this algorithm in both domains. Frequency domain smoothing is used for standard runs in this research.

Plotting Results

Plots of the errors in the algorithm's representation of the intensity functions and in tracking are presented in this section. For each setting of the parameter values of the previous section, 15 plots were generated. All of the plots are mean errors $\pm$ one standard deviation.

The first four plots are of the errors in the estimates of the x and y location of the target. Errors at a minus time are the errors before incorporation of a measurement, while errors at a plus time are after the measurement has been processed at that time. Plots five and six, of any 15 plot sequence, are the errors in x or y dynamic estimates at a minus time followed immediately by the errors at the plus time. Observing the way these plots are driven toward smaller errors at plus times relative to minus time estimates clearly shows how much information is obtained from that frame. Recall that the dynamics state estimates are needed
for the controller and errors in these estimates are the fundamental indicator of tracking performance, while centroid estimates are required for centering images in the upper processing path of Figure 1. Plots seven through twelve are the corresponding error plots for the centroid estimates. Plots 13 through 15 are plots of the intensity function spatial average derivation errors of Equations (6-3) and (6-4).

For each of the fifteen plots, a legend is provided to show how the variable parameters of the previous section were set for a given plot. This legend consists of the COV = $\sigma^2$, NZ = number of zero to pad, rows and columns used, ALP = $a$ used for smoothing, NF = number of frequencies to zero, VAB = variance of the background and FLIR noise, and SDF = sigma of the dynamics assumed by the filter. The following 15 plots, Figures 12 through 27, are provided here as an example of the sequence and results, for the case of COV=2, NZ=0, ALP=0.1, NF=0, VAB=1., SDF=.1. Appendix I contains plots for four other settings of the variable parameters.

Figures 12 and 13 show the x and y errors at a minus time. The plots indicate that the steady state error for either variable is reached by the fourth frame. The sigma is approximately constant after that point also. The same trend is also seen in the x and y plus time errors, Figures 14 and 15, except that the errors are smaller as expected after incorporation of a measurement. Plots 16 and 17 show
how much information is obtained, and the corresponding reduction in errors, by incorporating a measurement. For these plots the errors at a minus time are plotted just prior to their corresponding plus time errors. It is clear from these plots that consistently good information is being obtained from the noise-corrupted measurements under the standard parameter settings. Figures 18 through 23 provide similar plots for the centroid location estimates. The same trend is evident for these plots except that the errors are smaller. This result was expected in that it is easier to find centroids than it is to separate dynamic and atmospheric contributions to intensity function movement. Plots 24 through 27 present the intensity function derivation errors. These plots clearly show that mean errors are reduced very quickly to very small values; the standard deviations are also reduced accordingly. The inadequacy of the dynamics models used by the Kalman filter resulted in the propagated state estimates being consistently biased in the same direction in these plots. Since it was the main goal of this research to develop and implement tracking algorithms which derive the target intensity functions in an on-line manner, the adequacy of the dynamics model was not emphasized. This problem is readily rectified by a dynamics model more representative of the target characteristics than a simple first order Gauss-Markov position process in each direction. Moreover, the propagation errors also provide a means to determine how
Figure 13. Y Minus Errors
Figure 15. Y Plus Error
Figure 17. Y Position Error: $\tilde{y}_d(t_i^-)$ and $\tilde{y}_d(t_i^+)$ Errors
Figure 18. X Centroid Minus Error.
Figure 19. Y Centroid Minus Error
Figure 20. X Centroid Plus Error
Figure 21. Y Centroid Plus Error
Figure 22. X Centroid Position Error: Plus and Minus Errors
Figure 23. Y Centroid Position Error: Plus and Minus Errors
Figure 24. Error of Estimated $h$
Figure 25. Error of Estimated $H_x - \partial h/\partial x$
Figure 26. Error of Estimated $H_y = \Delta h/\Delta y$
much information is being obtained during measurement updates when these on-line derived intensity functions are being used.

Table Summary

This section will summarize the tracking and intensity function estimation abilities in tabular form. The plots of the previous section show errors plotted as a function of frame number. The results displayed in Tables I-III are the corresponding time average of those errors and their standard deviations over frames 10 to 20. The plots of the previous section implied that the steady state error region is reached by this time, and thus time averaging is confined to this interval. The special comment column is to designate pertinent information not indicated by the header (e.g. smooth in space, run made on Cyber 175/Modcomp minicomputer).

The first result from Table I is that exponential smoothing produces similar results in either the frequency or space domain. Entries two, four, seven, and nine of Table I are compared to the entries directly above them to show this result. This was expected from the derivation of exponential smoothing contained in Chapter II. There are additional errors induced when smoothing in space is used for this implementation. Recall Figure 1 of Chapter I. The upper path of that figure is where smoothing is accomplished. Before smoothing can be accomplished, the intensity function must be centered within the data frame. The centering is

<table>
<thead>
<tr>
<th>Table I</th>
<th>Table II</th>
<th>Table III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential smoothing</td>
<td>Similar results in both domains</td>
<td>Entries compared for relevant data</td>
</tr>
</tbody>
</table>

Run made on Cyber 175/Modcomp minicomputer.
Table I. Tracking and Intensity Function Derivation Errors for Modcomp (minicomputer) (32 bit)

Header = (COV,NZ,ALP,NF,VAB,SDT)

<table>
<thead>
<tr>
<th>Comments</th>
<th>Header</th>
<th>(\overline{X}<em>{\text{err}}(\cdot) / \overline{X}</em>{\text{err}}(-))</th>
<th>(\overline{X}<em>{\text{err}}(+)/ \overline{X}</em>{\text{err}}(\cdot))</th>
<th>(\overline{Cn}<em>{\text{err}}(\cdot) / \overline{Cn}</em>{\text{err}}(-))</th>
<th>(\overline{Cn}<em>{\text{err}}(+)/ \overline{Cn}</em>{\text{err}}(\cdot))</th>
<th>(\overline{h}<em>{\text{err}}/ \overline{h}</em>{\text{err}})</th>
<th>(\overline{h}<em>{\text{err}}/ \overline{h}</em>{\text{err}})</th>
<th>(\overline{h}<em>{\text{err}}/ \overline{h}</em>{\text{err}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard run</td>
<td>2,0,1,0,1..1</td>
<td>-1.4237/ .10685</td>
<td>-.033996/ .10559</td>
<td>-.13567/ .09443</td>
<td>-.02635/ .05220</td>
<td>-.00819/ .11294</td>
<td>.05159/ .16288</td>
<td>-.00441/ .15684</td>
</tr>
<tr>
<td>Smooth in Space</td>
<td>2,0,1,0,1..1</td>
<td>-1.4237/ .10684</td>
<td>-.03399/ .10559</td>
<td>-.13567/ .09443</td>
<td>-.02634/ .05221</td>
<td>-.00344/ .11294</td>
<td>.01748/ .16288</td>
<td>-.00445/ .15683</td>
</tr>
<tr>
<td>Cut 2 highest frequency</td>
<td>2,0,1,2,1..1</td>
<td>-1.3865/ .10813</td>
<td>-.03020/ .10697</td>
<td>-.12084/ .09441</td>
<td>-.02254/ .05344</td>
<td>-.00401/ .11172</td>
<td>.01492/ .12498</td>
<td>-.00096/ .12681</td>
</tr>
<tr>
<td>Smooth in Space</td>
<td>2,0,1,2,1..1</td>
<td>-1.3865/ .10813</td>
<td>-.03019/ .10697</td>
<td>-.12084/ .09441</td>
<td>-.02254/ .05344</td>
<td>-.00401/ .11172</td>
<td>.01491/ .12497</td>
<td>-.00099/ .12681</td>
</tr>
<tr>
<td>Cut 4 freq</td>
<td>2,0,1,4,1..1</td>
<td>-1.3062/ .10906</td>
<td>-.02181/ .10803</td>
<td>-.12392/ .09656</td>
<td>-.01415/ .05711</td>
<td>-.00650/ .10352</td>
<td>.00315/ .08090</td>
<td>-.00011/ .07722</td>
</tr>
<tr>
<td>Pad zeros</td>
<td>2,8,1,0,1..1</td>
<td>-1.5707/ .10600</td>
<td>-.05132/ .10505</td>
<td>-.15039/ .09612</td>
<td>-.04370/ .05658</td>
<td>-.00080/ .12000</td>
<td>.26469/ .12886</td>
<td>.66217/ .13602</td>
</tr>
<tr>
<td>Smooth Space</td>
<td>2,8,1,0,1..1</td>
<td>-1.5707/ .10600</td>
<td>-.05131/ .10505</td>
<td>-.15038/ .09612</td>
<td>-.04370/ .05658</td>
<td>-.00080/ .12000</td>
<td>.26468/ .12886</td>
<td>.66213/ .13602</td>
</tr>
<tr>
<td>Max Noise</td>
<td>2,0,1,0,4..1</td>
<td>-1.9813/ .11769</td>
<td>-.09482/ .11970</td>
<td>-.19151/ .11787</td>
<td>-.08732/ .08694</td>
<td>.00031/ .39491</td>
<td>.04328/ .62031</td>
<td>.02009/ .57596</td>
</tr>
<tr>
<td>Smooth Space</td>
<td>2,0,1,0,4..1</td>
<td>-1.9812/ .12059</td>
<td>-.09481/ .11970</td>
<td>-.19151/ .11787</td>
<td>-.08731/ .08694</td>
<td>.00031/ .39490</td>
<td>.04327/ .62031</td>
<td>.02006/ .57595</td>
</tr>
</tbody>
</table>
**Modcomp**

Header = (COV,NZ,ALP,NF,VAB,SDT)

<table>
<thead>
<tr>
<th>Comments</th>
<th>Header</th>
<th>(\bar{X}<em>{\text{err}}(-)/\sigma</em>{x_{\text{err}}}(+/-))</th>
<th>(\bar{X}<em>{\text{err}}(+)/\sigma</em>{x_{\text{err}}}(+/-))</th>
<th>(\bar{Cn}<em>{\text{err}}(-)/\sigma</em>{c_{\text{err}}}(+/-))</th>
<th>(\bar{Cn}<em>{\text{err}}(+)/\sigma</em>{c_{\text{err}}}(+/-))</th>
<th>(h_{\text{err}}/\sigma_{h_{\text{err}}}(+/-))</th>
<th>(\partial h/\partial x_{\text{err}}/\sigma_{x_{\text{err}}}(+/-))</th>
<th>(\partial h/\partial y_{\text{err}}/\sigma_{y_{\text{err}}}(+/-))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha = .2)</td>
<td>2,0,2,0,1,1,1</td>
<td>-1.4984/1.0516</td>
<td>1.13069/1.0397</td>
<td>-1.14316/1.09382</td>
<td>-0.03526/0.05135</td>
<td>0.00386/0.15074</td>
<td>0.02400/0.24566</td>
<td>0.00548/0.23296</td>
</tr>
<tr>
<td>(\alpha = .2)</td>
<td>2,8,2,0,1,1,1</td>
<td>-1.6290/1.0445</td>
<td>-0.05827/1.0345</td>
<td>-1.15622/0.09544</td>
<td>-0.05067/0.05561</td>
<td>0.00490/0.16088</td>
<td>0.27491/0.19251</td>
<td>0.66310/0.19765</td>
</tr>
<tr>
<td>Pad zeros</td>
<td>2,0,2,0,1,1,1</td>
<td>-1.6290/1.0445</td>
<td>-0.05827/1.0345</td>
<td>-1.15622/0.09544</td>
<td>-0.05067/0.05561</td>
<td>0.00490/0.16088</td>
<td>0.27491/0.19251</td>
<td>0.66310/0.19765</td>
</tr>
<tr>
<td>Perfect Shift</td>
<td>2,0,1,0,1,1,1</td>
<td>-1.6290/1.0445</td>
<td>-0.05827/1.0345</td>
<td>-1.15622/0.09544</td>
<td>-0.05067/0.05561</td>
<td>0.00490/0.16088</td>
<td>0.27491/0.19251</td>
<td>0.66310/0.19765</td>
</tr>
<tr>
<td>Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benchmark</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table II. Tracking and Intensity Function Derivation Errors for Cyber

<table>
<thead>
<tr>
<th>Comments</th>
<th>Header</th>
<th>$\bar{X}/\overline{\sigma_X}$</th>
<th>$\bar{Y}/\overline{\sigma_Y}$</th>
<th>$\bar{C}_n/\overline{\sigma_C}$</th>
<th>$\bar{C}_m/\overline{\sigma_M}$</th>
<th>$\bar{H}/\overline{\sigma_H}$</th>
<th>$\bar{H}/\overline{\sigma_X}$</th>
<th>$\bar{H}/\overline{\sigma_Y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard run</td>
<td>2,0,1,0,1,0,1,1</td>
<td>-.16401/0.09851</td>
<td>-.06009/0.10155</td>
<td>-.14510/0.09683</td>
<td>-.03973/0.05396</td>
<td>-.01362/0.12596</td>
<td>.01171/0.16935</td>
<td>-.00240/0.16071</td>
</tr>
<tr>
<td>Cut 2 highest freq</td>
<td>2,0,1,2,1,0,1,1</td>
<td>-.16133/0.09892</td>
<td>-.05716/0.10179</td>
<td>-.14246/0.09684</td>
<td>-.03682/0.05313</td>
<td>-.01593/0.11595</td>
<td>.00765/0.12728</td>
<td>.00080/0.12772</td>
</tr>
<tr>
<td>Cut 4 highest freq</td>
<td>2,0,1,4,1,0,1,1</td>
<td>-.15310/0.09985</td>
<td>-.04853/0.10316</td>
<td>-.13398/0.09763</td>
<td>-.02795/0.05532</td>
<td>-.01875/0.10966</td>
<td>.00285/0.08395</td>
<td>.00526/0.08908</td>
</tr>
<tr>
<td>Pad zeros</td>
<td>2,8,1,0,1,0,1,1</td>
<td>-.18205/0.08699</td>
<td>-.08080/0.09138</td>
<td>-.16392/0.09083</td>
<td>-.06079/0.05181</td>
<td>-.01355/0.14139</td>
<td>.25773/0.14166</td>
<td>.65931/0.15111</td>
</tr>
<tr>
<td>Pad zeros Cut 2 freq</td>
<td>2,8,1,2,1,0,1,1</td>
<td>-.15440/0.08918</td>
<td>-.05040/0.09328</td>
<td>-.13539/0.09126</td>
<td>-.02931/0.04822</td>
<td>-.05099/0.10413</td>
<td>.27265/0.10348</td>
<td>.34126/0.10350</td>
</tr>
<tr>
<td>Pad zeros Cut 4 freq</td>
<td>2,8,1,4,1,0,1,1</td>
<td>-.15217/0.08814</td>
<td>-.04815/0.09227</td>
<td>-.13309/0.09116</td>
<td>-.02762/0.04804</td>
<td>-.09408/0.08726</td>
<td>.26509/0.06571</td>
<td>.28585/0.05512</td>
</tr>
<tr>
<td>Max Noise</td>
<td>2,0,1,0,4,0,1,1</td>
<td>-.25885/0.12018</td>
<td>-.15837/0.12461</td>
<td>-.24168/0.12601</td>
<td>-.14068/0.10081</td>
<td>-.01382/0.48062</td>
<td>.04630/1.05467</td>
<td>.05107/0.95636</td>
</tr>
<tr>
<td>Max Noise Cut 2 freq</td>
<td>2,0,1,2,0,4,1,1</td>
<td>-.25601/0.11867</td>
<td>-.15474/0.12271</td>
<td>-.23904/0.09363</td>
<td>-.13636/0.09363</td>
<td>-.01692/0.45704</td>
<td>.03043/0.78877</td>
<td>.03624/0.72683</td>
</tr>
<tr>
<td>Max Noise Cut 4 freq</td>
<td>2,0,1,4,0,4,1,1</td>
<td>-.23200/0.12373</td>
<td>-.12861/0.12980</td>
<td>-.10967/0.10119</td>
<td>-.10967/0.10119</td>
<td>-.02014/0.46506</td>
<td>.00930/0.47468</td>
<td>.03069/0.45459</td>
</tr>
<tr>
<td>Set broad gaussians $\sigma^2 = 3$</td>
<td>( \frac{\bar{X}<em>{\text{err}}}{\sigma</em>{x\text{err}}} )</td>
<td>( \frac{\bar{X}<em>{\text{err}}}{\sigma</em>{x\text{err}}} )</td>
<td>( \frac{\bar{C}<em>{\text{err}}}{\sigma</em>{c\text{err}}} )</td>
<td>( \frac{\bar{C}<em>{\text{err}}}{\sigma</em>{c\text{err}}} )</td>
<td>( \frac{\bar{h}<em>{\text{err}}}{\sigma</em>{h\text{err}}} )</td>
<td>( \frac{\partial h/\partial x_{\text{err}}}{\sigma_{x\text{err}}} )</td>
<td>( \frac{\partial h/\partial y_{\text{err}}}{\sigma_{y\text{err}}} )</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>Broad gaussian pad zeros</td>
<td>3.8, 1, 0, 0, 0, 0, 0</td>
<td>-0.25443/0.07859</td>
<td>-0.15775/0.08227</td>
<td>-0.23708/0.09335</td>
<td>-0.13904/0.06827</td>
<td>-0.02478/0.18966</td>
<td>-0.96095/1.13722</td>
<td>1.12159/1.5447</td>
</tr>
<tr>
<td>Sharply peaked gaussian $\sigma^2 = 1$, max noise</td>
<td>1.0, 1, 0, 0, 0, 0, 0</td>
<td>-0.20193/0.10531</td>
<td>-0.09843/0.10804</td>
<td>-0.18386/0.17194</td>
<td>-0.07891/0.07194</td>
<td>-0.03004/0.38217</td>
<td>-0.11787/0.95185</td>
<td>0.91628</td>
</tr>
<tr>
<td>Sharply peaked max noise pad errors</td>
<td>1.8, 1, 0, 0, 0, 0, 0</td>
<td>-0.18863/0.10873</td>
<td>-0.08510/0.11151</td>
<td>-0.17026/0.10690</td>
<td>-0.06527/0.07131</td>
<td>-0.01977/0.37866</td>
<td>-0.07142/0.54349</td>
<td>0.16721/0.51137</td>
</tr>
<tr>
<td>Sharply peaked max noise pad zeros cut 4 freq</td>
<td>1.8, 1, 4, 0, 0, 0, 0</td>
<td>-0.16987/0.11241</td>
<td>-0.05443/0.11556</td>
<td>-0.15109/0.10979</td>
<td>-0.03619/0.07453</td>
<td>-0.01376/0.29974</td>
<td>-0.01352/0.31264</td>
<td>0.10246/0.26368</td>
</tr>
<tr>
<td>Sharply peaked $\alpha = 0.2$</td>
<td>1.0, 0, 2, 0, 0, 0, 0</td>
<td>-0.15465/0.09155</td>
<td>-0.05060/0.09321</td>
<td>-0.13560/0.08959</td>
<td>-0.03009/0.03813</td>
<td>-0.00430/0.13314</td>
<td>-0.02005/0.24674</td>
<td>-0.00978/0.22716</td>
</tr>
</tbody>
</table>
## Cyber

**Header**: (COV,NZ,ALP,NF,VAB,SDT)

<table>
<thead>
<tr>
<th>α = 0.2</th>
<th>2,0,2,0,1,1,1</th>
<th>$\bar{x}<em>{y</em>{\text{err}}}(-)/\sigma_{x_{\text{err}}}(-)$</th>
<th>$\bar{x}<em>{y</em>{\text{err}}}(+)/\sigma_{x_{\text{err}}}(+)$</th>
<th>$\bar{c}<em>{x</em>{\text{err}}}(-)/\sigma_{c_{\text{err}}}(-)$</th>
<th>$\bar{c}<em>{y</em>{\text{err}}}(+)/\sigma_{c_{\text{err}}}(+)$</th>
<th>$\bar{h}<em>{\text{err}}/\sigma</em>{x_{\text{err}}}$</th>
<th>$\partial h/\partial x_{\text{err}}/\sigma_x_{\text{err}}$</th>
<th>$\partial h/\partial y_{\text{err}}/\sigma_y_{\text{err}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>α = 0.2</td>
<td>2,0,2,0,4,1,1</td>
<td>$-0.17092/-0.09758$</td>
<td>$-0.06822/-0.10061$</td>
<td>$-0.15211/-0.09663$</td>
<td>$-0.04797/-0.05433$</td>
<td>$-0.01267/-0.16021$</td>
<td>$0.01559/-0.25261$</td>
<td>$-0.00094/-0.22793$</td>
</tr>
<tr>
<td>α = 0.2</td>
<td>2,0,2,0,4,1,1</td>
<td>$-0.15695/-0.09995$</td>
<td>$-0.05310/-0.10328$</td>
<td>$-0.13789/-0.09764$</td>
<td>$-0.03258/-0.05590$</td>
<td>$-0.01798/-0.13531$</td>
<td>$0.00014/-0.11204$</td>
<td>$0.00460/-0.11618$</td>
</tr>
<tr>
<td>α = 0.2</td>
<td>max noise</td>
<td>$-0.23271/-0.11564$</td>
<td>$-0.18662/-0.12021$</td>
<td>$-0.26580/-0.12513$</td>
<td>$-0.16843/-0.10165$</td>
<td>$-0.00989/-0.63206$</td>
<td>$0.08033/1.56522$</td>
<td>$0.09211/1.43047$</td>
</tr>
<tr>
<td>No smoothing</td>
<td>2,0,1,0,1,1,1</td>
<td>$-0.45490/-0.09643$</td>
<td>$-0.38392/-0.10125$</td>
<td>$-0.44009/-0.10821$</td>
<td>$-0.36811/-0.08998$</td>
<td>$-0.03799/-1.134$</td>
<td>$-0.01214/-1.7261$</td>
<td>$0.12642/-1.5972$</td>
</tr>
<tr>
<td>No smoothing</td>
<td>2,0,1,0,1,1,1</td>
<td>$-0.28138/-0.10730$</td>
<td>$-0.19140/-0.11404$</td>
<td>$-0.26407/-0.10679$</td>
<td>$-0.17283/-0.08334$</td>
<td>$-0.018096/0.78588$</td>
<td>$0.004024/0.68377$</td>
<td>$0.024496/0.68377$</td>
</tr>
<tr>
<td>Comments</td>
<td>Standard run</td>
<td>Pad zeros</td>
<td>Cut 2 highest frequency</td>
<td>Cut 4 highest frequency</td>
<td>Max Noise Cut 2 freq</td>
<td>Max Noise Cut 4 freq</td>
<td>Alpha = .2</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>-------------------------</td>
<td>-------------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>2.0,1,0,1,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,0,1,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
<tr>
<td>2.0,1,2,1,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,2,1,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
<tr>
<td>2.0,1,4,1,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,4,1,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
<tr>
<td>2.0,1,0,4,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,0,4,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
<tr>
<td>2.0,1,2,4,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,2,4,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
<tr>
<td>2.0,1,4,4,...,1</td>
<td>1.3668</td>
<td>-2.0829</td>
<td>-1.3610</td>
<td>-1.3630</td>
<td>-1.3512</td>
<td>-1.3512</td>
<td>-1.3982</td>
<td></td>
</tr>
<tr>
<td>2.8,1,4,4,...,1</td>
<td>-1.8451</td>
<td>-1.2413</td>
<td>-1.2850</td>
<td>-1.2440</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td>-1.2850</td>
<td></td>
</tr>
</tbody>
</table>

Table III. Tracking Errors for Correlator-Kalman Filter

Header = (COV,NZ,ALP,NF,VA3,SDT)
accomplished in the frequency domain, (see Chapter II). If smoothing is to be accomplished in the space domain, the centered data must then be inverse transformed. The smoothed data would then be transformed again for evaluation at the propagated state estimate and differentiation which are accomplished in the frequency domain. The additional transforms required for smoothing in space induce additional numerical errors. The very slight difference between the results shows that transforming errors are practically negligible. Smoothing in space may be used for the optical implementations which will be presented in the next chapter.

Table I contains the results obtained on the Modcomp while Table II contains the results obtained on the Cyber 175. The Modcomp minicomputer uses 32 bit arithmetic compared to the Cyber's 60 bits. The first entry from either table is the standard run as defined by the parameter settings contained in the header. The comparable results obtained in tracking errors for example in $\dot{x}_{\text{err}}(+) = -0.03399 \pm 0.10559$ compared to $-0.06009 \pm 0.10155$, reveal an insensitivity of the developed algorithms to word length. Similarly, comparable results were obtained for filtering out high frequencies, padding with zeros, and even the maximum noise corruption case. This result is important since it implies an insensitivity to $h$, $H_x$, and $H_y$ errors. The insensitivity also implies that $h$ and the Kalman filter gain, which uses $H_x$ and
H, could be refreshed at a lower cycle rate than the 30 Hz frame cycle time.

Zeroing out the two highest spatial frequencies is seen to increase accuracy on all columns in Table I. This was expected for the spread variable, \( \sigma^2 \), equal to two. (This will be explored further when broad and sharp intensity shapes are discussed below.) The values for the three Gaussian dispersions, whose peaks are separated by approximately four pixels, results in an intensity function which contains relatively low spatial frequencies. The high spatial frequency components are because of noise and the zeroing of them decreases the intensity function derivation errors and thereby also increases tracking accuracy. Tables I and II show that zeroing out the four highest spatial frequencies didn't result in a significant change in the errors from the two frequency cut off case.

Padding with zeros slightly increases errors over the pad with data case for the standard parameter settings. This shows that the true intensity profile is not equal to zero outside the 8 x 8 tracking window for the spread parameter equal to 2. To pad with zeros introduces artificial high frequencies as explained earlier, and zeroing out these artificial high frequencies improved tracking by 18 percent for \( \hat{x}(-) \) (see Table II entries 5 and 6), where only a 2 percent improvement was achieved when padding with data (see Table II
entries 1 and 2). For this application the luxury of padding with data exists because a frame of FLIR data actually consists of $300 \times 400$ pixels. Padding with zeros will be shown to enhance tracking for sharp intensity profile cases below. For sharp target intensity profiles the true target intensity is approximately zero outside the $8 \times 8$, so only noise contributes significant amplitudes there. Zeroing that noise enhances tracking.

The eighth entry of Table I provides the tracking and intensity function derivation errors when SNR is at its minimum level used in this research, i.e., SNR = 10. The increased noise did degrade tracking, by about 40% for $\hat{x}(-)$ and by about 300% for $\hat{x}(+)$, as was expected since it becomes harder to find the target within the data frame. The derivation errors for $h$, for the maximum noise corruption case of Table II entry seven, is seen not to change significantly from the case of high standard SNR as in entry one. This result shows that the pattern recognition algorithm is not very sensitive to the increase in noise.

Entries four, seven, and nine of Table I present information on the combination of smoothing in space with cutting the two highest spatial frequencies, padding with zeros or in combination with maximum noise corruption. Smoothing in space for all of these cases produced results compatible with smoothing in the frequency domain. This is seen when
each of the smoothing in space entries are compared with the entries directly above them in Table I.

Entry ten of Table I and entry sixteen of Table II result from using an alpha equal to .2 instead of the standard .1. Tracking ability is degraded from that of entry one even though errors in the derivation of \( h \) and \( H_y \) are not increased. However, errors in deriving \( H_k \) do increase. Since setting alpha equal to .2 simulates a finite memory filter of about five samples long, this result shows that a higher magnitude steady state error level is reached for \( H_k \) because less smoothing is accomplished. But, if the target shape does change, this larger alpha would respond faster than the smaller standard alpha. Recall that the placement of the three Gaussians, as explained in Chapter V, is the reason that \( H_k \) is harder to derive than \( H_y \). The no smoothing entry of Table II shows that no smoothing increases tracking errors, \( \hat{x} (+) \), by almost 400%. Cutting four frequencies reduces tracking errors to about 300% of the smoothed error.

The sharply peaked intensity entries of Table II show that tracking and intensity function derivation errors increase over the corresponding broad intensity cases. The increased errors are because the target's intensity functions are not as easily distinguished from the noise as when the target has a broad intensity profile. Padding with zeros for the sharply peaked intensity entries of Table II decreases
tracking errors by 10%. Recall that padding with zeros increased errors for the $\sigma_g^2 = 2$ case. The placement of the Gaussians with small dispersions result in the target only contributing insignificant magnitudes outside the 8 x 8 tracking window. The padding with zeros only essentially zeros only noise for these cases resulting in smaller errors.

Last it should be noted that the Correlator-Kalman filter algorithm of Chapter V outperformed the Figure 1 algorithm consistently. The first entry of Table III shows that for the standard case that the correlator-Kalman filter algorithm has smaller tracking errors, 25% less for $\hat{x}(+)$, and centroid location errors, 32% less for $\hat{x}_{\text{peak}}(+)$. The sensitivity of this algorithm to the variation of parameters is similar to the results discussed above. Padding with zeros increased errors, by 35% for $\hat{x}(-)$, while cutting high frequencies did not have much effect on the standard case. Maximum noise increases errors, by 20% for $\hat{x}(+)$, but cutting high frequencies from this minimum SNR case regains 9% of that lost accuracy. Setting alpha equal to .2, entry 8 Table III, is found not to induce additional tracking errors, unlike the effect it had on the algorithm of Figure 1. Since this algorithm does not use $H_x$, this result was expected. Also it must be re-emphasized that this is not a standard correlator. This correlation algorithm uses dynamic modelling information from the Kalman filter to position the template.
Thresholding is used to avoid false peaks along with increasing the accuracy of the centroid calculation. Thresholds of .1 and .9 were tested for no noise and for SNRs of 10 and 20. The mean error, for SNR = 10, decreased from .08211 pixels for no threshold to -.03870 pixels for a threshold of .2. Although these mean errors are both close to zero and their difference may be considered insignificant, the standard deviation made a significant decrease from .01357 to .00531 pixels. At a threshold of .5 the mean error had increased to .080 pixels and the standard deviation had dropped to .00363 pixels. The histogram of the errors also approximated the Gaussian form more than the .2 or .3 threshold cases. This Gaussian form along with the tighter standard deviation led to the choice of using a threshold of .5 for this research.

The third enhancement of the correlator is that the template is being derived on-line via the data processing of Chapter II. Finally, the enhanced correlator position estimates are processed by a Kalman filter. Capt Mercier's research (5) tested a standard correlation tracker against an extended Kalman filter algorithm which utilized apriori knowledge of the intensity functions. In Table II of that study (5:57) the mean track error, for a standard correlation tracker, is listed at .5 pixels with a 1σ error of 1.5 pixels for a SNR = 20. The correlation algorithm developed for this research resulted in a mean error of -.02543 pixels with a 1σ
error of .13437 pixels. This algorithm shows considerable potential as a next generation tracker.

Summary

This chapter presented the results from the testing of the tracking algorithms developed in the previous chapters. The errors are plotted as mean ± one sigma (standard deviation) errors and tables of errors are generated for variable settings for Kalman filter and pattern recognition parameters. Extremely good tracking is established for both algorithms with the Correlator-Kalman filter algorithm showing considerable potential.

The better performance of the alternate tracking algorithm can be explained as follows. Previous comparisons (4; 5) of Kalman filter trackers to correlators used apriori knowledge of the intensity functions. As less knowledge is available, the processing of the 64 intensity measurements via a Kalman filter is less accurate, with less relative benefit over performance attainable with a correlator. Moreover, as discussed in the previous section, this alternate tracker embodies much more than a simple correlation tracker.
VII. Optical Processing Alternatives

Introduction

The computational burden of either of the tracking algorithms developed in the previous chapters is heavy. For that reason, a serious attempt needs to be made to develop hybrid, optical and digital, systems for implementation of those algorithms. Optical processing offers the potential of parallel processing of two-dimensional data. The two-dimensional Fourier transforms, correlations, and matrix operations which are contained in the tracking algorithms could potentially be accomplished at extremely high data rates using optical techniques. A digital computer could be programmed to perform necessary data analysis or control the optical computations. Such a hybrid system would combine the speed and parallel processing advantages of optical processors with the flexibility of digital computers.

This chapter begins with a section which describes the basic principles of optical processing and introduces the concept of hybrid processing. The next section adapts the hybrid processor to the tracking algorithms developed previously. Potential modifications to the tracking algorithms are proposed to make optical implementation easier. The purpose of this section is to present design possibilities and to provide an extremely rough, preliminary analysis of the potential for fruition of optical implementations for
each proposal.

**Background**

This section presents the basic principles of optical processing systems. These systems utilize optical interference to process two-dimensional incoming signals. The most common system used for optical processing is shown in Figure 27. Light from a laser point source $S$ is collimated by lens $L_c$. The collimated beam of quasi-monochromatic coherent laser light is used to illuminate the object in plane $P_1$ and a diffraction pattern is formed. The object is placed one focal length in front of the transforming lens $L_1$. The classical Fraunhofer diffraction pattern of the object's space-varying amplitude transmittance, $t_0(x_1,y_1)$, in plane $P_1$ is formed in the rear focal plane of lens $L_1$. The complex field of the Fraunhofer diffraction pattern in plane $P_2$ is mathematically equal to the spatial Fourier transform (7:166) of the object.

$$T_0(x_2,y_2) = k_1 \int t_0(x_1,y_1) \exp\left[-\frac{j2\pi}{\lambda f_1}(x_2x_1 + y_2y_1)\right] dx_1 dy_1 \quad (7-1)$$

$$T_0(f_x,f_y) = k_1 \int t_0(x_1,y_1) \exp\left[-j2\pi(f_xx_1 + f_yy_1)\right] dx_1 dy_1 \quad (7-2)$$

where

$k_1 =$ complex constant

$T_0 =$ Fourier transform of $t_0(x_1,y_1)$

$\lambda =$ mean wavelength of the laser
The two-dimensional Fourier transform of the object in plane \( P_1 \) can be modified by placing an appropriate optical filter in plane \( P_2 \). This optical filter is in general a complex function. The optical filter diffracts the light incident upon plane \( P_1 \), producing a second diffraction pattern (14:28). This diffraction pattern is then focussed by lens \( L_2 \) onto plane \( P_3 \).

In a hybrid system, the input plane \( P_1 \), the transform plane \( P_2 \), and the output image plane \( P_3 \) are all interfaced via spatial light modulators to a digital computer. To utilize the processing capability of the optical processor, some means of inputting information into and reading information out of these spatial light modulators at data rates commensurate with the 30Hz FLIR frame repetition rate is required. Spatial light modulators with cycle frequencies of 30Hz are now available from several vendors.

Itek Corporation, for example, produces a Pockels Read-out Optical Modulator, (PROM). The PROM is a spatial light modulator which is constructed from bismuth silicon oxide (bisox). Bisox is a crystal which displays both wavelength selective photoconductivity and the linear electro-optic (Pockels) effect (15:353). The complexity of the total exposure-readout process has precluded the development of a complete and exact theoretical model of PROM operation for predicting system response for varying operating conditions.
However, simplified models which are limited to particular phases of the process have been developed. These theoretical models include predictions on the wavelength dependence of PROM sensitivity, the relationship between PROM sensitivity and Bisox crystal thickness, and how PROM resolution can be increased by increasing capacitance of the blocking layer (15:364). The characteristics of the PROM, or of any of the spatial light modulators considered in this analysis do not lend themselves to presentation in a simple tabular form because of the many variables involved in the complex exposure-readout process. However, certain benchmarks on commercially available PROMs are available. A 25mm diameter is standard with special order devices with 37mm diameters also available. The desired 30Hz cycle rate is standard with a dc contrast ratio of $10^3$ to $10^4$:1. A spatial frequency response of 100 cycles per millimeter has been demonstrated (15:364).

Hughes also produces a spatial light modulator which could be used in this application. The hybrid field-effect liquid crystal light value (LCLV) is a high resolution optical-to-optical image converter. The input and output light beams are completely separated and noninteracting. The parameters which characterize LCLV performance are: sensitometry; modulation transfer function and resolution; linearity; signal-to-noise ratio; response time; optical flatness; and
image quality (16:374). The performance of the LCLV can only be given for specific values of the operational parameters (voltage bias and frequency, and orientation of the light value with respect to the output light polarization direction). The performance is particularly sensitive to the imaging light power. The use of intensifier tubes which are capable of amplifying very low light level scenes to the levels required by the light value is feasible (16:381).

These are not the only available spatial light modulators. Their characteristics are however, indicative of currently available devices.

Applying Optical Processing to Tracking

This section will apply the optical processing techniques discussed in the previous section to the two tracking algorithms developed in chapters one through six. This section will also propose potential modifications to the tracking algorithms to facilitate hybrid implementations.

The data processing algorithm of Chapter II is first considered for optical implementation. Both tracking algorithms use this procedure for generating information about the target's intensity profile. In addition, the algorithm of Figure 1 also uses the derivative information developed. The software implementation of the pattern recognition algorithm, Appendix H, along with the explanation of Chapter II clearly shows how manipulation of the amplitudes and phases
of the various spatial frequencies are used to derive the respective intensity functions. A complex filter placed in the transform plane, $P_{2}'$, of Figure 27 can be used to manipulate the amplitudes and phases of those spatial frequencies simultaneously. The spatial filtering results of the previous chapter become important here. Complex filters are usually implemented using holographic techniques which were pioneered by A. B. Vander Lugt of the University of Michigan's Radar Laboratory. The best hybrid combination for generation of the flexible complex filter which would accomplish the application of the appropriate linear phase shift to center the target's intensity function within the incoming data array requires further research. Once this filter or combination of filters is synthesized and centering of the data is accomplished, using the configuration of Figure 27 with the complex filter discussed above in plane $P_2$, plane $P_3$ would contain the space domain representation of the centered target intensity function. The next step is to smooth this image with the previous best estimate of the intensity function stored in a spatial light modulator. Since exponentially smoothing can be accomplished in either domain, as established in Chapter VI, there is no constraint on where an optical implementation performs it. Both images are multiplied by constants (see Chapter II) and the products are added, probably via simultaneous recording techniques, to
form the new best estimate of the intensity function which is written into the spatial light modulator which stores the smoothed estimate. This smoothed estimate must also be processed by a complex shifting filter to produce the intensity function expected at the next sample time, \( h(\hat{\mathbf{v}}(t_i), t_i) \). Differentiation of this expected intensity profile can be accomplished using a simple amplitude filter in the transform plane, \( P_2 \) of Figure 27. The derivative of the object, in plane \( P_1 \) of Figure 27 can be produced by a filter function whose amplitude transmittance is

\[
f(k_x) = a(k_0 - k_x)
\]

where

\[ a, k_0 = \text{constants} \]

The use of the bias term \( k_0 \) eliminates the need for a complex filter (15:361) for differentiation. This amplitude transmittance filter has its maximum transmittance at the center of the field-of-view with a minimum transmission on the edges. This is analogous to using the differentiation property of the Fourier transform as explained in Chapter II. Digitally controlled complex filters could theoretically accomplish the Chapter II intensity function derivations. Such filters are not easily implemented. The information which is to be input into the transform plane filter spatial light modulator would be computed by a digital computer. The computed desired transfer function is added to the
reference wave within the digital computer to simulate the holographic synthesis technique. The output of the digital computations is a non-negative, real-valued function which is quantized. Since the desired masks are of finite extent, recording of sampled values of the computed transfer function can completely specify its Fourier transform. This assumes that the well known sampling theorem (7:25) is satisfied.

The major advantage of digitally constructed filters is that an abstract, mathematically defined filter can be computed and recorded. This eliminates the need for the availability of the point-spread function. Such complex digital computations could minimize the advantage of a hybrid implementation for a small tracking window such as the 8 x 8 pixels used in this research. However, if this intensity function could be shown to be only slowly changing a slower repetition rate on its generation could relieve the digital computational burden.

Another possible application of optical processing to the algorithm of Figure 1 would be in the implementation of the high dimensioned matrix operations. The parallel nature of optical processing could be used to great advantage in accomplishing these functional operations. The formation of the expected intensity function, $h(\hat{x}(t_i), t_i)$, was discussed above. Once the spatial domain representation of $h(\hat{x}(t_i), t_i)$ is created, it is to be subtracted from the measurement data $z(t_i)$, which is the incoming FLIR data frame. The intensity
values from the FLIR could drive a spatial light modulator and then the residual could be formed by optical subtraction of $z(t_i) - h(x(t_i), t_i)$ (16:383). The subtraction scheme of Reference 16, page 383, uses two LCLVs onto which $z(t_i)$ and $h(x(t_i), t_i)$ are projected. Potentially the IR radiation which was input to the FLIR could directly drive a spatial light modulator eliminating the need for a scanning type FLIR. The residual must next be multiplied by the gain matrix $K(t_i)$. In the digital implementation this matrix is $4 \times 64$ with only two unique rows. Therefore, two two-dimensional images which represent the two unique rows could be placed, via spatial light modulators, in the plane, in the space domain, which contains the residual resulting in two-dimensional images which could be sampled and summed to produce the two unique update components of Equation (4-2). These are the components which are added to our best estimate of the state before we get a measurement, $\hat{x}(t_i^{-})$, to generate $\hat{x}(t_i^{+})$.

To enhance the algorithm of Chapter V, Correlator-Kalman filter, optical correlations can be used. The transform of the template will be encoded into the filter of the transform plane, $P_2$, of Figure 27. This template could theoretically be obtained by accomplishing the data processing of Chapter II optically as discussed above. A second possibility would be to derive the template digitally as was done for Chapter V and then input the information into the spatial
light modulator; this was also discussed above. A third possibility would be to array many potential templates in the filter plane and accomplish simultaneous optical correlations with all of them. Templates could be prepared in which the presence and disposition of targets were varied and where the background characteristics were also varied. The weighting of the results from the multiple template correlations could be controlled via adaptive estimation of target and environmental characteristics. The input data image would be used to modulate a collimated laser beam which is input to a transform lens. A multiple holographic lens is used as the Fourier transforming lens which directs the transform to each of the many matched filters, made from the templates. The position estimates from any of these correlation implementation options could then be processed by a Kalman filter.

Conclusion

In summary, high bandwidth optical processing in parallel with flexible lower bandwidth digital computation potentially could relieve the computational burden of the tracking algorithms developed in this research. The pattern recognition algorithm given in Chapter II would be very difficult to implement optically because of the complex filters required. For implementation of the high dimensional Kalman filter, (see Chapter IV), optical processing could accomplish the computationally intense functional matrix operations.
Optical implementation of the correlation algorithm of Chapter V has the most promise. Before any optical implementation is possible though, considerable research and development must be accomplished. The software developed in this research provides a flexible tool to determine what spatial frequency modifications enhance tracking. The zeroing out of spatial frequencies within the software is just one example of how this software can help determine what filters are needed for optical implementation of the tracking algorithms and what characteristics the resulting hybrid optical/digital algorithm would have. The testing of the differentiation filter of Equation (7-3) would just be a point by point multiplication of the spatial frequency components by an array whose values were determined by that equation. The software provided in Appendix H is well commented and easily modified for this type of testing.
VIII. Conclusions and Recommendations

Conclusions

The conclusions addressed in this section are formulated from the details of Chapter VI, Performance Analysis. The milestones which were realized in this research will be reiterated here.

As shown in Table I, the extended Kalman filter algorithm performs very well even when the intensity functions are not assumed, as was done in prior research efforts (4;5). These intensity functions were derived using the digital pattern recognition techniques developed in Chapter II. It was shown in Chapter VI that the tracking capability was not overly sensitive to increased errors in the derivation of the intensity functions. The filtering of high spatial frequencies was shown to decrease tracking errors for cases where the intensity function is relatively smooth.

In previous research efforts, apriori intensity function information was given to the extended Kalman filter. For some tracking scenarios those extended Kalman filter trackers were found to outperform correlation trackers. The alternate tracking algorithm of Chapter V was developed to combine correlation and Kalman filtering. When no apriori information on the intensity functions was assumed, the processing of the 64 intensity measurements by the Kalman filter should not outperform correlators so easily. For this reason
a correlation algorithm which makes use of dynamic model information from the Kalman filter, thresholding correlator outputs to remove false peaks prior to centroid calculations, on-line template derivation and correlator position estimate enhancement via a Kalman filter was derived. Chapter VI showed how this algorithm outperforms the originally envisioned algorithm of Figure 1.

Finally Chapter VII presented the optical processing alternatives for implementing the tracking algorithms. The parallel nature of optical processors provide the potential to relieve the computational burden of the tracking algorithms. Prior to any optical implementation, extended research and development must be accomplished.

Recommendations

Further research is recommended to assess the effects of incorporating the IR laser's FLIR image in the model's used. The use of optical windows could potentially notch the contribution of the IR laser out of the FLIR image. The laser will however induce hot spots on the target which were not part of the original multi-hotspot target model. Modern estimation techniques could also be used to determine what translational or phase distortion effects occurred during propagation of the laser beam. The use of spatial light modulators could potentially obtain the phase distortion induced by the atmosphere. A compensating imposed distortion on the
outgoing beam could then maximize target damage.

Another area of suggested research is to attempt optical implementation of those portions of the tracking algorithms discussed in Chapter VII. This must include research into the feasibility of using the many types of available spatial light modulators.

The alternative tracking algorithm developed in Chapter V needs to be investigated further. The optimal thresholding and peak detection or centroiding techniques could further improve tracking ability. Adaptive techniques for on-line tuning could also be investigated.

The algorithms of this research should now be tested using real FLIR data. A combined Physics-EE effort at AFIT would be required to assemble the experimental apparatus needed to generate the real FLIR data. As an alternative, the data could potentially be provided to AFIT by the Air Force Weapons Laboratory.

The use of different dynamics models should also be investigated. Adaptive models or models compatible with different tracking scenarios could also be investigated.

An investigation of how these algorithms perform when tracking a slowly changing target shape is also needed. The variation of alpha to enhance tracking of slowly changing targets could also be investigated.

Finally, only empirical convergence of the algorithms
developed was shown during this research. A theoretical proof of convergence of these adaptive estimation techniques should be derived.
Bibliography


2. Robinson, Clarence A. Jr., "Beam-Target Interaction Tested," Aviation Week and Space Technology: 114, 14-17 (0'8'79).


Appendix A

Two-Dimensional Finite Discrete Fourier Transform
Interpretation and Test

Similar to the presentation of J. W. Goodman, reference 7, the Rect function is defined to extract one period of the spatial intensity function as

\[
\text{Rect}_N(x,y) = \begin{cases} 
1 & 0 \leq x < N-1, \ 0 \leq y < N-1 \\
0 & \text{otherwise}
\end{cases} \quad (A-1)
\]

Reproducing Equations (2-3) and (2-4) of Chapter II and using the Rect function to define the area sequence \(g(x,y)\) as being zero outside the interval \(0 \leq x < N-1\)

\[
g(x,y) = g'(x,y) \text{ Rect}_N(x,y) \quad (A-2)
\]

\[
G(f_x, f_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x,y) e^{-i2\pi(f_x x + f_y y)} \text{ Rect}_N(f_x, f_y) \quad (A-3)
\]

\[
g(x,y) = \frac{1}{N^2} \sum_{f_x = 0}^{N-1} \sum_{f_y = 0}^{N-1} G(f_x, f_y) e^{i2\pi(f_x x + f_y y)} \text{ Rect}_N(f_x, f_y) \quad (A-4)
\]

\(G(f_x, f_y)\) is the Finite Discrete Fourier Transform of \(g(x,y)\) (8:118). The Rect function of Equation (A-1) is separable in the two independent variables:

\[
\text{Rect}_N(f_x, f_y) = \text{Rect}_N(f_x) \text{ Rect}_N(f_y) \quad (A-5)
\]

Equation (A-3) can now be written as
\[ G(f_x, f_y) = \sum_{y=0}^{N-1} X(f_x, y) \exp \left( \frac{-12\pi i (f_y y)}{N} \right) \text{Rect}_N(f_y) \quad (A-6) \]

where
\[ X(f_x, y) = \sum_{x=0}^{N-1} g(x, y) \exp \left( \frac{-12\pi i (f_x x)}{N} \right) \text{Rect}_N(f_x) \quad (A-7) \]

Equation (A-7), \( X(f_x, y) \) corresponds to an \( N \)-point one dimensional Discrete Fourier Transform for each value of the row index \( y \) (8:118). \( X(f_x, y) \) is the result of \( N \) one dimensional transforms, one for each row of \( g(x, y) \). In Figure A-1a, if \( y \) is held constant, say \( y=2 \), and a one-dimensional Fourier Transform is accomplished across that row, the variation in the image intensities from column to column would result in the zero frequency of that variation to be located at \( x'=1 \) and \( y'=2 \), of Figure A-1b, while the coefficient associated with the fundamental, the first harmonic, in both directions would be found at \( x'=2 \) and \( y'=2 \) with its conjugate located at \( x'=N \) and \( y'=2 \).

To complete the two-dimensional Discrete Fourier Transform Equation (A-6) expresses how to implement the \( N \) one-dimensional transforms along each column. To summarize this general interpretation of the two-dimensional Discrete Fourier Transform, the Equations (A-6) and (A-7) show how the two-dimensional transform is achieved by using a one-dimensional transform on the rows first and then on the columns, or vice versa.
Figure A.1. Row Transform Result of 2-D DFT
For the case where the image intensity function is separable in the two independent variables \(x\) and \(y\), for example, \(g(x,y)\) having the property that

\[
g(x,y) = g_1(x)g_2(y)
\]  

(A-8)

the two-dimensional Discrete Fourier Transform, \(G(f_x,f_y)\), becomes the product of the one-dimensional independent transforms \(G_1(f_x)\) and \(G_2(f_y)\).

\[
G(f_x,f_y) = G_1(f_x)G_2(f_y)
\]  

(A-9)

A function of two independent variables is separable within a specific coordinate system if it can be written as a product of two functions of one independent variable each. The two-dimensional transform degenerates to holding the row index constant, for example, and running the one-dimensional transform across the columns, for anyone of the \(N\) rows. Similarly the column index is held while the one-dimensional transform is accomplished across the rows, and this is done for any of the columns. The results of these two independent transforms are then multiplied together.

This leads to some interesting algebra which can be exploited to test the implementation of the two-dimensional Discrete Fourier Transform. In Figure A-1, \(T_{(2,2)}\) will consist of the product of the two one-dimensional transform fundamental coefficients for that row or column. If \(x\) is the column coordinate and \(y\) is the row coordinate, then
\[ T_{(Y,x)} \] is defined as

\[
T_{(2,2)} = \left[ \text{1st harmonic in } y \right] \hat{\Delta}_y (1,2) \cdot \left[ \text{1st harmonic in } x \right] \hat{\Delta}_x (1,2) \\
(A-10)
\]

\[
T_{(2,N)} = \left[ \text{1st harmonic in } y \right] \hat{\Delta}_y (1,N) \cdot \left[ \text{conjugate of 1st harmonic in } x \right] \hat{\Delta}_x (1,2)^* \\
(A-11)
\]

\[
T_{(N-1,2)} = \left[ \text{conjugate of 1st harmonic in } y \right] \hat{\Delta}_y (1,2)^* \cdot \left[ \text{1st harmonic in } x \right] \hat{\Delta}_x (1,N-1) \\
(A-12)
\]

\[
T_{(N-1,N)} = \left[ \text{conjugate of 1st harmonic in } y \right] \hat{\Delta}_y (1,N)^* \cdot \left[ \text{conjugate of 1st harmonic in } x \right] \hat{\Delta}_x (1,N-1)^* \\
(A-13)
\]

if \( y_1 < N/2, x_1 < N/2 \)

\[
T_{(y_1,x_1)} = \left[ \text{(y\_1-1) harmonic in } y \right] \hat{\Delta}_y (y_1-1,x_1) \cdot \left[ \text{(x\_1-1) harmonic in } x \right] \hat{\Delta}_x (x_1-1,y_1) \\
(A-14)
\]

If the variation in intensity across every row is equal to the variation in intensity for every other row and the variation across all of the columns is similarly set equal, then the components of the transform can be defined as
\[ Y(1,2) = a + jb \quad x(1,2) = c + jd \]
\[ Y(1,N) = a + jb \quad x(1,N-1) = c + jd \]
\[ Y(1,2)^* = a - jb \quad x(1,2)^* = c - jd \]
\[ Y(1,N)^* = a - jb \quad x(1,N-1) = c - jd \]  \hspace{1cm} (A-14)

Using Equation (A-14) to solve (A-10) through (A-13)

\[ T(2,2) = Y(1,2)^* x(1,2) = (a+jb)(c+jd) = (ac-bd) + j(bc+ad) \]  \hspace{1cm} (A-15)

\[ T(2,N) = Y(1,N)^* x^*(1,2) = (a+jb)(c-jd) = (ac+bd) - j(ad-bc) \]  \hspace{1cm} (A-16)

\[ T(N-1,2) = Y^*(1,2) x(1,N-1) = (a-jb)(c+jd) = (ac+bd) + j(ad-bc) \]  \hspace{1cm} (A-17)

\[ T(N-1,N) = Y^*(1,N) x^*(1,N-1) = (a-jb)(c-jd) = (ac-bd) - j(ad+bc) \]  \hspace{1cm} (A-18)

Equation (A-15) for \( T(2,2) \) is the conjugate of Equation (A-18) for \( T(N-1,N) \) under the conditions of uniform variations on any row and similarly for the columns, which just implies separability.

To implement the two-dimensional Fourier Transform, subroutine Fourt was used, which is a common Fortran subroutine (4:). To test subroutine Fourt to see where it places harmonics with a two-dimensional array, the results of Equations (A-15) through (A-18) were used. Figures A-2 through A-7 show the results of the tests.

136
Figure A-4. (sin 2nx/6)(sin 2ny/24)
Figure A-5. \((\sin 2\pi x/4)(\sin 2\pi y/24)\)
Figure A-6. \((\cos \frac{2\pi x}{2})(\sin \frac{2\pi y}{24})\)
Figure A-1 had as its input a 24 x 24 array with magnitudes of each element varying only across the rows via $\cos(2\pi x/6)$. Using the above interpretation of a two-dimensional Fourier Transform, the fundamental frequency assumed would be one which corresponded to one period across the 24 element array. Since the input obviously fits 4 cycles in that space, the only nonzero component of the Fourier Transform expected would be at the fourth harmonic. The transform would also be expected to be real only since the input is a pure cosine. Figure A-2 is then encouraging in that the only nonzero components are where expected and the imaginary portions of that answer is extremely small. The reason that there are nonzero elements only in the first row is that when the transform was accomplished across the columns, there was no variation in the y coordinate, which is equivalent to taking the transform of a constant one. That transform results in a one in the zero frequency components in each column which is the first row and zeros everyplace else. When the multiplication of Equation (A-9) is accomplished, Figure A-2 results.

Figure A-3 had an input of $(\cos 2\pi x/6) \cdot (\cos 2\pi y/24)$ which is the same variation in the x-direction and a variation in the y-direction which corresponds to one period exactly. The only nonzero result from the transform along the columns, variation in y, should be in the fundamental, which it is.
Figure A-4 through A-7 are just further examples to demonstrate Fourt. Figure A-4 had an input of the fourth harmonic in the x-direction and the fundamental in the y-direction. Figure A-5 contains the sixth harmonic in the x-direction. Figure A-6 contains the twelfth harmonic in the x-direction which only appears as a conjugate. Figure A-7 contains only the twelfth harmonic in both directions and encloses spatial frequencies within boxes. A thorough understanding of these results is needed to understand taking derivatives and shifting in the transform domain.

In summary Figure A-8 is provided to show the output of the subroutine Fourt. The DC component is the product of the zero frequency components of the one-dimensional transforms in both directions. Elements $T_{(1,2)}$ through $T_{(24,24)}$ can be interpreted similarly to Equations (A-10) through (A-13).
Figure A-8. Output of Fourt

Zero frequency in y variation → DC $T(0,2)$

$11^{th}$ harmonic in y variation → $T(24,1)$

Conjugate $12^{th}$ harmonic in y

Conjugate $1^{st}$ harmonic in y

$11^{th}$ harmonic in x direction → $T(24,24)$

Conjugate $12^{th}$ harmonic in x

Conjugate $1^{st}$ harmonic in x
Appendix B

Shifting Property of the Two-Dimensional Finite Discrete Fourier Transform

To accomplish interframe filtering, the target's intensity profile must be centered within each data frame. This would allow the successively centered frames to be averaged to attenuate noise. However, for the reasons presented in Chapter I, the true target's intensity profile is offset from the center of the field-of-view. This offset from the center of the data array is estimated by the Kalman filter updated states, \( \hat{x}(t_i^+) \). The problem then becomes how to manipulate the data array so that the target's intensity profile is centered within the field-of-view. The solution is to use the shifting property of the Fourier Transform to alter the data to negate a spatial shift of an amount calculated from the filter's updated estimate.

As stated in Chapter II, a shift of a function in the space domain introduces a linear phase shift in the frequency domain (8:9). Equation (8) of Chapter II is now rewritten to include the summations of the Two-Dimensional Finite Discrete Fourier Transform.

\[
F \left[ g(x-x_0, y-y_0) \right] = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x,y) \exp \left[ -\frac{i2\pi}{N} (fx+fy) \right] \exp \left[ -\frac{i2\pi}{N} (fx_0+fy_0) \right] \\
0 \leq y_0 < N \quad 0 \leq x_0 < N
\]

(B-1)

146
This equation can be shortened, as done with Equation (8) of Chapter II by substituting $G(f_x', f_y')$ for the Fourier Transform of the unshifted function $g(x,y)$.

$$
F\left[ g(x-x_0', y-y_0') \right] = G(f_x', f_y') \exp \left[ \frac{-i2\pi}{N} (f_x'x_0 + f_y'y_0) \right] \\
0 < y_0' < N \\
0 < x_0' < N \quad \text{(B-2)}
$$

The reason for the qualifier of shift magnitudes under Equations (B-1) and (B-2) is that if a shift were allowed to be greater than the period assumed by the Fourier Transform, see Appendix A, it could not be distinguished from a shorter shift. This is because of aliasing. Therefore, $x_0 < N$ and $y_0 < N$ are required for the interpretation of the shift as a rotation in two dimensions to be a valid unique representation.

The implication of Equations (B-1) and (B-2) using different factors for different spatial frequencies is that the results of Appendix A, Two-Dimensional Fourier Transforms, must be used to locate each of the spatial frequencies within the transform domain. To implement the shift, each point in the transform domain must be multiplied by the conjugate of the phase shift induced by the spatial translation.

$$
g(x,y) = F^{-1}\left[ G(f_x', f_y') \exp \left[ \frac{-i2\pi}{N} (f_x'x_0 + f_y'y_0) \right] \exp \left[ \frac{+i2\pi}{N} (f_x'x_0 + f_y'y_0) \right] \right] \\
= F^{-1}\left[ F[g(x-x_0', y-y_0')] \exp \left[ \frac{+i2\pi}{N} (f_x'x_0 + f_y'y_0') \right] \right] \quad \text{(B-3)}
$$
The centered intensity function \( g(x,y) \) can be derived from the inverse Fourier Transform of the Fourier Transform of the shifted function \( F[g(x-x_0,y-y_0)] \) via Equation (B-3). The shifted intensity function \( g(x-x_0,y-y_0) \) is the input frame of data in Figure 1 of Chapter I. The Two-Dimensional Finite Discrete Fourier Transform of this data is then taken, which is \( F[g(x-x_0,y-y_0)] \). In Appendix A, Figure A-8, the output of a two-dimensional transform is shown. To implement a shift each point of the two-dimensional transformed data array must be multiplied by the conjugate of the translation induced linear phase shift as shown in (B-3).

Subroutine Shift shifts a data array by an amount specified by its parameters. In Figure A-8, for example, point \( T(1,2) \) is the point of the transformed array in the first row second column. Figure A-8 shows it is the first harmonic in the \( x \)-variation and the zero frequency in the \( y \)-variation (i.e., \( f_x = 1, f_y = 0 \)). This point would therefore be multiplied by \( \exp[-\frac{12\pi}{N}x_0] \) to implement a shift of \( x_0, y_0 \). Taking another point, for example, \( T(24,24) \) is the transformed data array component in the twenty-fourth row twenty-fourth column. Figure A-8 shows this point to be the product of the conjugate of the first harmonics in both directions. To implement a shift this point would be multiplied by \( \exp[\frac{+12\pi}{N}(x_0+y_0)] \).

Figure B-1a and B-1b show the results of subroutine Shift. The inputs to Shift are the data array, the dimension...
Figure B-1. Results of Subroutine Shift
of that data array, and the amount in pixels of shift desired. This routine was tested with data representing a single Gaussian and a multi-hotspot three Gaussian profile.

In summary, Equation (B-3) shows how the centered intensity function can be derived from the Fourier Transform of the spatially translated intensity function $F(g(x-x_0, y-y_0))$. The Two-Dimensional Finite Discrete Fourier Transform assumes that this finite-area array represents one period of a two-dimensional periodic sequence. For this reason, if a shift is greater than the period it can't be distinguished from a shorter shift.
Appendix C

Implementation of the Exponential Smoothing Algorithm

As explained in Chapter II, the target's intensity pattern is corrupted by noise. Interframe smoothing of centered intensity patterns is accomplished with an exponential smoothing algorithm defined by

\[
\hat{y}(t) = \alpha y(t) + (1-\alpha) \hat{y}(t-1)
\]  

where

\[
\hat{y}(t) = \text{current averaged data frame}
\]
\[
y(t) = \text{current data frame}
\]
\[
\hat{y}(t-1) = \text{previous result of averaging data}
\]
\[
\alpha = \text{smoothing constant} \quad 0 < \alpha < 1
\]

For smoothing in the frequency domain Equation (C-1) becomes

\[
\hat{L}(t_i) = \alpha \hat{L}(t_i) + (1-\alpha) \hat{L}(t_{i-1})
\]  

where

\[
\hat{L}(t_i) = \text{current averaged data frame in frequency domain}
\]
\[
\hat{L}(t_i) = \text{transform of current noise corrupted data frame}
\]
\[
\hat{L}(t_{i-1}) = \text{previous result of averaging in frequency domain}
\]
\[
\alpha = \text{smoothing constant} \quad 0 < \alpha < 1
\]

If, for example, the steady state value of \( \alpha \) is to be 0.1, this algorithm would be implemented as follows. The smoothing constant, \( \alpha \), is varied for the first ten frames.
(α=1/k k=1,2,3,...10) until the steady state value of α = .1 is reached. Once steady state is reached Equation (C-2) becomes:

\[ \hat{L}(t_i) = .1 \hat{L}(t_i) + .9 \hat{L}(t_{i-1}) \]  \hspace{1cm} (C-3)

when the smoothing constant, \( \alpha \), is set to 1/k for the first ten frames, k is equal to from one to ten, the equations below describe the smoothing algorithm.

\[
\begin{align*}
    k &= 1 \quad \hat{L}(t_1) = L(t_1) \\
    k &= 2 \quad \hat{L}(t_2) = \frac{1}{2}L(t_2) + \frac{1}{2}L(t_1) \\
    k &= 3 \quad \hat{L}(t_3) = \frac{1}{3}L(t_3) + \frac{2}{3}L(t_2) = \frac{1}{3}L(t_3) + \frac{2}{3}[\frac{1}{2}L(t_2) + \frac{1}{2}L(t_1)] \\
    & \quad \quad \quad = \frac{1}{3}L(t_3) + \frac{1}{3}[L(t_2) + L(t_1)] \\
    k &= 4 \quad \hat{L}(t_4) = \frac{1}{4}L(t_4) + \frac{3}{4}L(t_3) \\
    & \quad \quad \quad = \frac{1}{4}L(t_4) + \frac{3}{4}[\frac{1}{3}L(t_3) + L(t_2) + L(t_1)] \\
    & \quad \quad \quad \quad = \frac{1}{4}[L(t_4) + L(t_3) + L(t_2) + L(t_1)] \hspace{1cm} (C-4)
\end{align*}
\]

This time averaging of the first k frames continues until the steady state value of \( \alpha = .1 \) and Equation (C-3) is reached. For the tenth frame the smoothing algorithm becomes:

\[
\begin{align*}
    k &= 10 \quad \hat{L}(t_{10}) = \frac{1}{10}L(t_{10}) + \frac{9}{10}[\frac{1}{9}L(t_9) + \ldots + L(T_1)] \\
    & \quad \quad \quad = \frac{1}{10}[L(t_{10}) + \ldots + L(t_1)]
\end{align*}
\]
For $k > 11$ the exponential smoothing equation can be generalized to the following equation

$$L_{k-11} = \hat{L}(t_{k-11}) = \frac{1}{10} L(t_{11}) + \sum_{i=1}^{10} (.9)(.1) L(t_i) + (.9)(.9)(.1) \sum_{i=1}^{10} L(t_i)$$

When $k = 11$ the second term of Equation (C-6), the summation from $i = 1$ to $k - 11$, becomes zero since the upper limit on the summation is less than the lower limit.

If it is desired to do this algorithm in the spatial domain the same summations hold.

$$h(x, y, t_i) = .1 h(x, y, t_{i-1}) + \sum_{k=1}^{i} (.9)(.1) h(x, y, t_{i-k}) + (.9)^{i-10} (.1) [h(x, y, t_{10}) + \ldots + h(x, y, t_1)]$$

In summary, an exponential smoothing algorithm was used to minimize the effects of the corrupting noise. Equations (C-4), (C-5), and (C-6) express the implementation of
This algorithm in the frequency domain. This same algorithm is valid in the spatial domain as shown in Equation (C-7).
The extended Kalman filter algorithm requires the spatial derivatives of the two-dimensional intensity function with respect to each of the states (see Chapter IV). The derivative property of the Fourier Transform can be used to obtain these derivatives.

\[ F\left[ \frac{\partial h(x,y)}{\partial x} \right] = j2\pi f_x \cdot F[h(x,y)] \]

\[ F\left[ \frac{\partial h(x,y)}{\partial y} \right] = j2\pi f_y \cdot F[h(x,y)] \]  

Equation (D-1) shows that differentiation in the x direction of the space domain is equivalent to multiplication by \( j2\pi f_x \) in the frequency domain and similarly for the y direction. The derivative of the intensity function can be expressed in terms of the transform of that function.

To implement Equation (D-1), the results of Appendix A must be used to locate each of the spatial frequencies in the transform domain. To obtain the spatial derivative of the intensity function, each data point in the transform domain must be multiplied by \( j2\pi \cdot \text{(corresponding spatial frequency in the direction of the desired derivative)} \). In Figure A-8, for example point \( T_{(1,2)} \) is the point of the transformed array in the first row and second column. Figure A-8 shows this point to be the product of the first harmonic
in the x-variation and the zero frequency in the y-variation. To obtain the spatial derivative in the x-spatial-direction, this point would be multiplied by \((j2\pi/N)\). Taking another point, say \(T(24,24)\), which is the transformed data array component in the twenty-fourth row and twenty-fourth column, the conjugate of spatial frequencies can be demonstrated. \(T(24,24)\) is the product of the conjugate of the first harmonics in both x and y directions. To obtain the spatial derivative in the x-direction this point would be multiplied by \(-j2\pi/N\).

To show how the Derivative algorithm was tested Figures D-1 through D-3 are included. Figure D-1 shows the original data to be sinusoids in both spatial directions, of one period in x and two periods in y, \(\sin(2\pi x/24) \cdot \sin(2\pi y/12)\). Figure D-2 shows the result of using the Derivative Property of the Fourier Transform to obtain the spatial derivative in the x-direction. The result is a cosinusoid of one period. Similarly, Figure D-3 shows the spatial derivative in the y-direction. The result is a cosinusoid of two periods. These figures use a grey-scale routine which attempts to show increases in intensity by more over-prints. (see Sub-routine Display). This routine only reflects the absolute values of intensities.

In summary, Equation D-1 shows how the spatial derivative can be expressed in terms of the transform of that in-
Figure D-1. Original Data $\sin \frac{2\pi}{24} x \cdot \sin \frac{2\pi}{24} y$
Figure D-2. Spatial Derivative in x
Figure D-3. Spatial Derivative in $y$
tensity function. Within the computer simulation, subroutine Deriv implements this algorithm. This appendix derives the spatial derivative of an intensity function. As shown in Chapter IV the algorithm will require the negative of this result.
Appendix E

Generation of White Gaussian Noise Process

Throughout this research, it was desired to generate samples of a discrete-time white Gaussian noise vector process with a mean of zero and a given variance. This function must be simulated through the use of pseudorandom codes that generate numbers as though they were generated as samples of a scalar random variable with uniform probability density between 0 and 1. (see Figure E-1)

![Figure E-1. Output Probability Density of Pseudorandom Code](image)

The mean and variance of this distribution is given by

\[
\mu_X = \int_{-\infty}^{\infty} \lambda f_X(\lambda) d\lambda = \int_{0}^{1} \lambda d\lambda = \frac{1}{2} \quad (E-1)
\]

\[
\sigma_X^2 = \int_{-\infty}^{\infty} (\lambda - \frac{1}{2})^2 f_X(\lambda) d\lambda = \int_{0}^{1} (\lambda^2 - \lambda + \frac{1}{4}) d\lambda = \frac{1}{12} \quad (E-2)
\]

If twelve independent calls are made to such a random number
generator and the outputs of these calls, specific realizations $\xi_i$, are summed, as

$$\Omega_j = \sum_{i=1}^{12} \xi_i$$

the result is a realization of a random variable with a mean of 6 and a variance of 1 that is essentially Gaussian (by the Central Limit Theorem, Law of Large Numbers, etc.). Six is then subtracted from each sum of realizations, $\Omega_j$, to produce a discrete realization of $\xi_j$ which will have a Gaussian distribution of zero mean and unity variance

$$\xi_j = \Omega_j - 6$$

Now the $\xi_j$'s are arrayed into a vector of appropriate dimension. For example to create a 64 x 1 vector $w_1$, 64 realizations of $\xi_j$, each being created from independent calls to the pseudorandom code, would be arrayed:

$$w_1 = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_{64} \end{bmatrix}$$

The vector process $w_1$ is composed of independent scalar white Gaussian noises of mean zero and unit variance.
To create a discrete-time white Gaussian noise vector process \( w_d(t_i) \), described by mean of zero and covariance of \( Q_d(t_i) \), in general non-diagonal, the following equation can be used.

\[
    w_d(t_i) = \sqrt[\text{Cholesky}] {Q_d(t_i)} \ w_1(t_i)
\]

where \( \sqrt[\text{Cholesky}] \) denotes Cholesky lower triangular square root. (13:408) Reference 13 shows that Equation (E-6) properly models the desired characteristics.

In summary, this appendix shows how samples of discrete-time white Gaussian noise vectors were generated from pseudo-random computer codes.
Appendix F

Discrete Representation of the Derivative of the Intensity Profile with Respect to the Kalman Filter States

To present a clearer presentation of the discrete representation of the intensity profile and its derivative with respect to the Kalman filter states, the continuous one-dimensional Gaussian profile of Figure F-1a is used. If the discrete representation of this intensity function is approximated as a single value at the center of a pixel the Figure F-1b is appropriate.

At time $t_1$, assume that the intensity profile is centered in the field-of-view of eight horizontal pixels, as in Figure F-1b. Although $h$ is actually the averaged value over a pixel it is approximated as the sampled value at the center. At time $t_2$, the target has moved in the direction of increasing state by an amount $x_0$, see Figure F-2. Taking the first pixel as an example, $h_1$ decreased in magnitude as the state increased. The magnitudes of the intensity measurements for all pixels located at state values to the left of the profile's maximum decreased in intensity while those to the right increased in magnitude. This is the spatial derivative rotated 180 degrees about the vertical axis. Figure F-3 gives the spatial derivative of a one-dimensional intensity profile as in Figure F-1, while Figure F-4 gives the derivative with respect to the Kalman filter states.
In summary, the derivative of the intensity profile with respect to the Kalman filter states is the negative of the spatial derivative of that intensity profile with respect to coordinate directions.
Figure F-1a. Continuous One-Dimensional Gaussian

Figure F-1b. Discrete One-Dimensional Gaussian
Figure F-2. Shifted Intensity Profile

Figure F-3. Intensity's Spatial Derivative
Figure F-4. Derivative with Respect to States

\[ \frac{\partial h}{\partial X_{\text{states}}} \]

\(x_{\text{direction}}\)
Appendix G
Monte Carlo Study

For the Monte Carlo analysis, many samples of the error process are generated within the computer simulation, and the error statistics are computed from those samples. In this research, twenty FLIR data frames represent one tracking history in time. For each quantity of interest, errors are computed before and after processing a FLIR data frame. These critical values, which the Kalman filter estimates, are compared to the truth model output to compute errors, as depicted in Figure G-1 (13:327).

![Diagram](image)

Figure G-1. Generation of Error Samples

169
The truth model generates the measurement vector $z_t(t_i)$ which is a realization of the measurement process $z_t(t_i, \omega_k)$ at time $t_i$, i.e. $z_t(t_i, \omega_k)$. The feedback loop only controls the center of the field-of-view for which the intensity measurements are generated. Twenty such twenty-frame time histories are now processed and the statistics of the errors, at each time frame are computed by averaging across the twenty time histories. For example say the difference between the truth model value for $x$ dynamics, $x_{td}$, and the filter estimated value for $x$ dynamics, $\hat{x}_d$, in the first time history, at frame one was $e_{1lx_d}^l$. By keeping stored $e_{ikx_d}^l$ and $e_{ikx_d}^2$ for each frame $i$ in a given simulation run, $i=1, \ldots, 20$, and for the twenty time histories $k=1, 2, \ldots, 20$, the mean error and the variance of the errors for any frame can be computed.

Figure G-1 assumes that the true values of the critical quantities $y_t$ are related to the truth model states, $x_t$, by a linear transformation represented by the matrix $C_t$. The matrix $C$ is similarly explained. For every quantity of critical interest, the appropriate entries in the 'C' matrices designate what combination of states constitute that value so the error sample can be generated.

In summary, the object of this Monte Carlo study was to characterize the error process of the algorithm of Figure 1 statistically.
Appendix H

Computer Software

This appendix contains the Fortran source code for some of the computer programs written in this study. The first program contained in this appendix was written for use on the CDC Fortran IV compiler. This program is the completed implementation of the algorithm of Figure 1. The next program was written for use on the MODCOMP Classic minicomputer to implement the algorithm of Chapter V. The last listing is the program which was written to generate the plots of Chapter VI. This software is well commented to allow for ease of understanding and modification.
PROGRAM MAIN (INPUT, OUTPUT, TAPES=INPUT, TAPES=OUTPUT, TAPES="",
1 DEBUG=OUTPUT)

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

C
C ** COMMENTS ON DATA STRUCTURES
C
C******************************************************************************

C ** IMAX IS AN ARRAY WHICH CONTAINS PEAK VALUES FOR THE THREE GAUSSIANS
C ** RMS BACKGROUND DIVIDED INTO THIS MAX VALUE QUAL SNR
C **
C ** S IS AN ARRAY WHICH CONTAINS INVERSE COVARIANCES OF STATES
C **
C ** XMAX IS A REAL ARRAY WHICH CONTAINS LOCATION OF MAX X COOODOF 3 GAUSS
C **
C ** YMAX IS SIMILAR TO XMAX
C **
C ** R IS THE CORRELATION MATRIX FILLED BY SUBROUTINE SPIN
C ** IT CONTAINS THE SPATIAL NOISE CORRELATION COEFFICIENT
C ** USING FIRST AND SECOND NEAREST NEIGHBOR
C ** REAL IMAX(3), S(12), XMAX(3), YMAX(3), R(54, 54)
C **
C ** XTREAL ARRAY OF TRUTH MODEL STATES
C ** SEE SUB TRUTH
C **
C ** PHIT - TRUTH MODEL STATE TRANSITION MATRIX SEE SUB TRUTH
C ** QD - QD - CHOLESKY SQUARE ROOT OF QD*XG(QD)
C ** YMAT WILL MULTI GO BY YU CORRUPT THE TRUTH MODEL STATES PG 15 MERCIER
C **
C ** H - THE MATRIX WHICH DETERMINES THE OUTPUT COMBINATION OF STATES
C **
C ** YT - OUTPUT EQUATIONS
C ** REAL XT(3,1), PHIT(3,3), QDROOT(3,3), WT(4,1), H(2,8), YT(2,1)
C **
C ** ROT-- ROOT OF R
C **
C ** W - VECTOR OF INDEPENDENT GAUSS N(0, 1)
C **
C ** WILL MULTI BY REJCT TO GET STATE UNCERTAINTY NOIS CORRUPTION
C **
C ** V-ROOT-W
C **
C ** UC - FOR OUTSIDE TRACKING WINDOW NEED SPATIALLY UNCORRELATED 24X24
C ** REAL PHAT(64, 64), XT(64, 64), UC(576)
C **
C ** DATE- ERROR IN RECONSTRUCTION OF DATA ARRAY, ONE NUMBER PER PIXEL PER FRAME
C **
C ** DEX- ERROR IN FORMATION OF DERIVATIVE WITH RESPECT TO X
C **
C ** DXE- SEE DEX
C **
C ** REAL DATE(8, 20), DEX(8, 20), DXE(8, 20)
C **
C ** DATE- SEE SUB SIMILAR TO DXE, DEX
C ** REAL DATE(8, 20), DEXE(8, 20), DXE(8, 20)
C **
C ** N1 IS AN ARRAY WHICH CONTAINS DIMENSION OF DATA SEE FOUNT
ENHANCED TRACKING OF AIRBORNE TARGETS USING FORWARD LOOKING INF--ETC(U)

S K ROGERS

AFIT/GEO/EE/BID-5
**PROGRAM MAIN**  
74/74  OPT=1  PMOMP  
FTN 4.8+52B  09/21/81  12.51  PAGE 2

```
INTEGER NN(2)
C** DATA- COMPLEX ARRAY OF DATA TO USED BY FOURT
C** WORK- SEE COMMENTS IN FOURT
C** DX, Dy, -DERIVATIVES
C** COMPLEX DATA(24,24), WORK(50), DX(24,24), Dy(24,24)
C** SD-PERFECT SHIFT/MANIPULATED DATA-SAVE DATA
C** SX, SY-SAVE PERFECT D/DX, D/DY
C** SDATA- SMOOTHED DATA
C** COMPLEX SD(24,24), SX(24,24), SY(24,24), SDATA(24,24)
C** FILTERS DATA STRUCTURES

PHIF IS THE STATE TRANSITION MATRIX FOR THE KALMAN FILTER
C** SEE SUBROUTINE FILTER
QFD IS THE RESULT OF THE INTEGRAL TERM IN THE PROPAGATION
C** OF THE COV MATRIX SEE SUBROUTINE PROP
PFM IS THE FILTERS COVARIANCE MATRIX PLUS- AFTER INCORPORATION
C** OF A MEASUREMENT
PFM IS THE FILTERS COVARIANCE MATRIX MINUS AFTER PROPAGATION
C** BUT PRIOR TO MEASUREMENT INCORPORATION
XP IS THE FILTER STATE VECTOR PLUS
XFM IS THE FILTER STATE VECTOR MINUS

REAL PHIF(4,4), QFD(4,4), PFM(4,4), XFM(4), XFM(4)
C** RINV IS THE INVERSE OF R WHICH IS NEEDED FOR THE PROPAGATION
C** OF THE INVERSE COVARIANCE METHOD

REAL RINV(64,64)
C** Z IS THE KALMAN FILTER MEASUREMENT VECTOR

REAL Z(64)
C** LINH IS AN ARRAY WHICH IS NEEDED IN THE EXTENDED KALMAN FILTER
C** THE PARTIAL OF THE INTENSITY FUNCTION WITH RESPECT TO
C** FILTERS BEST ESTIMATE OF STATES
C** NLINH IS AN ARRAY WHICH IS FILLED FROM THE SMOOTHED DATA AND IS
C** THE NONLINEAR INTENSITY FUNCTION WHICH IS TO BE USED TO
C** PROCESS THE NEXT MEASUREMENT

REAL LINH(64,4), NLINH(64)
C** SAVE IS A TEMPORARY USE MATRIX IN CONJUNCTION WITH SDATA
C** COMPLEX SAVE(24,24)
```
DATA STRUCTURES TO GATHER STATISTICS ON FILTER TRACKER CAPABILITY

XFME IS THE ERROR BETWEEN THE PREDICTED X DYNAMIC LOCATION AT A PARTICULAR MINUS TIME AND THE TRUTH MODEL TRUE X DYNAMIC LOCATION

XFME2 IS THE SQUARE OF THE XFME

NOTE THAT XFME AND XFME2 ARE ARRAYS WHICH ARE DIMENSIONED TO BE 2X20 THE FIRST ROW IN EACH IS USED FOR THE X DIRECTION WHILE THE SECOND ROW IS FOR THE Y DIRECTION

CNME IS THE ERROR IN THE PREDICTED LOCATION OF THE CENTROID AT A PARTICULAR MINUS TIME COMPARED TO THE TRUTH MODEL

CNME2 IS THE SQUARE OF CNME

NOTE AGAIN THE DIMENSION OF CNME AND CNME2C
REAL XFME(2,20),XFME2(2,20),CNME(2,20),CNME2(2,20)

XFPE IS THE ERROR BETWEEN THE UPDATED DYNAMIC LOCATION AT A PARTICULAR PLUS TIME AND THE TRUTH MODEL TRUE DYNAMIC

XFPE2 IS THE SQUARE OF XFPE

NOTE**
AFTER CALL TO SUBROUTINE FILST THE SQUARED MATRICES ARE SET TO STANDARD DEVIATIONS

NOTE THE DIMENSIONALITY OF XFPE,XFPE2 FOR THE SAME REASONS AS ABOVE THE 20 ALLOWS COMPUTATION OF STATISTICS PER FRAME UP TO 20 FRAME

CNPE IS THE CENTROID ERROR AT THE PLUS TIME
CNPE2C
REAL XFPE(2,20),XFPE2(2,20),CNPE(2,20),CNPE2(2,20)

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

INITIALIZATION OF DATA STRUCTURES

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

INITIALIZE TARGET INTENSITY ASSUMING 3 CIRCULAR COSS SECTION GAUSS TARGET

DATA IMAX/20.,20.,20./
DATA S/1.000.,0.,0.,1.000.,0.,0.,1.000.,0.,0.,1.000./
DATA XMAX/0.,-2.,2./
DATA YMAX/-2.666666.,1.333333.,1.333333./

X,Y ARE THE CORNERS OF THE FOV THEY ARE SET HERE TO CENTER THE CENTROID FOR THE GIVEN INITIAL CONDITIONS ON XMAX,YMAX

DATA X/-4./
DATA Y/-4./
DATA NN/24.,24./

TDIS IS THE CORRELATION TIME FOR THE GAUSS MARKOV KALMAN FILTER
DATA TDF/1.5/

******************************************************************************
IFLAG WILL DETERMINE IF ANOTHER SET OF VARIATIONS TO THE INPUT NEEDS TO BE PROCESSED

3985 IFLAG=0

SETUP IS THE INITIALIZATION ROUTINE
CALL SETUP(COV, S, NZ, NZM, NFAMES, NRUNS, ALPHA, NFREQ, 1ISF, IEF, VARM, SIGDT, DT, TD, IFLAG, PHIT, QDROOT, H, TAF, 3VARDF, VARAF, TDF, PHIF, QPD, DATE, DATE2, DEX, DEX2, 3DYE, DYE2, XFMX, XFXME2, CNME, CNME2, XFXPE, XFXPE2, 4CNPE, CNPE2, NMAX, XMAX, YMAX, X, Y, S, Sx, SY, R, RROOT, RINV)
IF(IFLAG.NE.0) GO TO 6421

BEGIN MONTE CARLO SIMULATION LOOP

DO 90 NS=1,NRUNS
XSHIFT=0.0
YSHIFT=0.0
C INITIALIZE SMOOTH DATA
DO 7 I=1,24
DO 7 J=1,24
7 SDATA(I,J)=CMPLX(0.,0.)

ZERO INITIAL CONDITIONS FOR THE FILTER

DO 106 I=1,4
DO 106 J=1,4
PFM(I,J)=0.0
PFP(I,J)=0.0
XFP(I)=0.0
106 XFM(I)=0.0
PFP(1,1)=.0375
PFP(2,2)=.0375
PFP(3,3)=.0025
PFP(4,4)=.0025
C INITIAL CONDITIONS ON DYNAMIC STATES
XFP(1)=3.0
XFP(2)=3.0
XFM(1)=XFP(1)
XFM(2)=XFP(2)

INITIALIZE STATE VECTOR
DO 71 I=1,8
71  \texttt{XT(1,1)=0.}

\texttt{YT(1,1) \text{ IS THE XPEAK OF THE CENTROID}}
\texttt{YT(2,1) \text{ IS THE YPEAK OF THE CENTROID}}
\texttt{XT(1,1)=3.0}
\texttt{XT(2,1)=3.0}
\texttt{YT(1,1)=3.0}
\texttt{YT(2,1)=3.0}

\texttt{DEFINE UPPER LEFT CORNER OF FOV}
\texttt{CENTERS CENTROID IN THE FOV}
\texttt{X=XFP(1)-4.}
\texttt{Y=XFP(2)-4.}

\texttt{TRACK TARGET FOR NFRAME FRAMES (TIME SLICES)}
\texttt{DO 90 NR=1,NFRAMES}

\texttt{DEFINE GAUS PEAK LOCATIONS BASED ON CENTROID POSITION,YT}
\texttt{XMAX(1)=YT(1,1)}
\texttt{XMAX(2)=YT(1,1)-2.}
\texttt{XMAX(3)=YT(1,1)+2.}
\texttt{YMAX(1)=YT(2,1)-2.66666}
\texttt{YMAX(2)=YT(2,1)+1.333333}
\texttt{YMAX(3)=YT(2,1)-1.333333}

\texttt{WRITE(6,410)}
\texttt{FORMAT('///,* X TRUTH*,4X,* Y TRUTH*,4X,* X FOV*,4X,* Y FOV*)}
\texttt{WRITE(6,411) YT(1,1),YT(2,1),X,Y}

\texttt{WRITE(6,174) NR,YT(1,1),YT(2,1),X,Y,XFM(1),XFM(2),XT(1,1),XT(2,1)}
\texttt{FORMAT(T2,*FRAME*,4X,T12,*XPOS*,E10.3,T27,*YPOS*,E10.3,T42,}
\texttt{*XFOV*,E10.3,T57,*YFOV*,E10.3,T12,*XFM*,E12.5,T30,*YFM*,}
\texttt{E12.5,*,T12,*XT*,E12.5,T30,*YT*,E12.5)}

\texttt{GET MEASUREMENT NOISE ARRAY}
\texttt{CALL NOISE(W,64)}
\texttt{CALL MULTI(RROOT,W,64,64,1,V)}

\texttt{GET MEASUREMENT DATA}
\texttt{CALL INPUT3(IMAX,S,XMAX,YMAX,24,X,Y,DATA,CENX,CENY)}
\texttt{CALL IDEAL3(IMAX,S,XMAX,YMAX,24,NZ,X,Y,DATA,DX,DY)}
\texttt{IF(NR.EQ.1) CALL DISPLAY(24,24,DATA)}
\texttt{IF(NR.EQ.1) CALL DISPLAY(24,24,DX)}
\texttt{IF(NR.EQ.1) CALL DISPLAY(24,24,DY)}

\texttt{ADD CORRELATED MEASUREMENT NOISE AT CENTER ,KB}
\texttt{DO 4 I=1,8}
DO 4 J=1,8
DATA(I*8, J*8)=DATA(I*8, J*8)+CMPLX(V(8*(I-1)+J),0,0)
        4 CONTINUE

290 ADD Uncorrelated noise TO measurement data outside center

        CALL NOISE(U,576)
        DO 6 I=1,24
        DO 6 J=1,24
        DATA(I,J)=DATA(I,J)+CMPLX((24*(I-1)+J),0.)*VARM
        CONTINUE

300 GET FORWARD FFT

305 CREATE THE MEASUREMENT VECTOR FOR THE FILTER UPDATE

310 K=0
            DO 101 I=9,16
            DO 101 J=9,16
            K=K+1
            Z(K)=REAL(DATA(I,J))

101 GO CALCULATE THE ERRORS OF THE FILTERS ESTIMATE PRIOR TO
      MEASUREMENT INCORPORATION

320 CALL STATFM(XFME,XFME2,CNME,CNME2,xfm,xt,yt,nr)

325 INCORPORATE MEASUREMENT

325 IF(NR.EQ.1) GO TO 164
            CALL UPDATE(2,linh,nlinh,xfp,xfm,pfp,pm,ri

330 CALCULATE THE ERRORS FOR THE FILTER AFTER THE INCORPORATION

330 OF THE MEASUREMENT
            CALL STATFP(XFPE,XFPE2,CNPE,CNPPE2,xt,yt,xfp,nr)

335 COMPUTE THE SHIFT INFORMATION FROM THE CENTER OF FOV
            XSHIFT=X-XFP(1)+4.-XFP(3)
            YSHIFT=Y-XFP(2)+4.-XFP(4)

340 WRITE(6,173)XSHIFT,YSHIFT

173 FORMAT(12,*XSHIFT=*,E10.3,T42,*YSHIFT=*,E10.3)

340 CALL FOURT(DATA,NN,2,-1,1,WORK)

340 FILTER desired frequency
IF(NFREQ.GT.12) NFREQ=12
IF(NFREQ.LE.0) GO TO 3796

DO 8 I=1,16
DATA(I,J)=CMPLX(0.,0.)
8 CONTINUE

IF(NR.NE.1) CALL SHIFT(DATA,24,XSHIFT,YSHIFT)
CALL SMOOTH(DATA,DATA,ALPHA,24,NR)

CALL DISPLAY(24,24,SX)
CALL DISPLAY(24,24,SDATA)
CALL DISPLAY(24,24,DX)
CALL DISPLAY(24,24,ST)
CALL DISPLAY(24,24,DY)

CALL PROPF(PHIF,QFD,PFM,FPM,FPM)
X=XM(1)=4.
Y=XM(2)=4.
XSHIFT=XM(3)
YSHIFT=XM(4)

ROUTINE FOR COMPUTING THE PERFECT FOR ERROR COMPUTATION
CALL PERF(X,Y,XSHIFT,YSHIFT,SX,SY,IMAX,XMAX,YMAX,S,NZ)

IF DESIRE PERFECT INTENSITY FUNCTION DATA UNCOMMENT NEXT CALL TO
TO PERF AND COMMENT 14 LINES FOLLOWING IT

CALL PERF(X,Y,XSHIFT,YSHIFT,SAVE,DX,DY,IMAX,XMAX,YMAX,S,NZ)

SAVE(I,J)=SDATA(I,J)
CALL SHIFT(SAVE,24,XSHIFT,YSHIFT)
CALL DERIV(SAVE,24,DX,DY)
CALL FOUNT(SAVE,NN,2,1,1,WORK)
CALL FOUNT(DX,NN,2,1,1,WORK)
CALL FOUNT(DY,NN,2,1,1,WORK)

SCALE THE INVERSE TRANSFORM ALONG WITH SETTING THE APPROPRIATE
SIGN TO INDICATE THE CHANGE IN INTENSITY PATTERN WITH RESPECT TO A
CHANGE IN STATE

DO 172 I=1,12
DO 172 J=1,24
DX(I,J)=DX(I,J)/576.
SAVE(I,J)=SAVE(I,J)/576.
DY(I,J)=DY(I,J)/576.
172 FILL IN THE LINEARIZED INTENSITY ARRAY

K=0
DO 102 I=9,16
DO 102 J=9,16
K=K+1
LINH(K,1)=REAL(DX(I,J))
LINH(K,3)=LINH(K,1)
LINH(K,2)=REAL(DY(I,J))
LINH(K,4)=REAL(SAVE(I,J))
LINH(K,5)=LINH(K,2)

102

ACCUMULATE ERRORS IN INTENSITY FUNCTIONS

CALL ACCUM(SUMDATA,SAVE,SD,SUMDX,DX,SUMDY,DY,SY,
#DATE,DATE2,DXE,DXE2,DYE,DYE2,NR)

PROPAGATE TRUTH MODEL STATE ONE FRAME

CALL PROP(PHIT,QDROOT,H,XT,YT,8,2)

90
CONTINUE

END MONTE CARLO SIMULATION LOOP

END MONTE CARLO SIMULATION LOOP

CALCULATE MEAN AND VARIANCE STATISTICS

GO CALCULATE THE FILTER STATISTICS

CALL FILST(XFME, XFME2,CNME,CNME2, XFPE, XFPE2, CNPE, CNPE2, NRUNS,
#NFRAMES)

CALL STAT(NFRAMES, DATE, NRUNS, DYE, DXE, DATE2, DXE2, DYE2, NZ, NFREQ, COV,
#VAR, ALPHA, XSHIFT, YSHIFT, SIGDT)
GO TO 3985

STOP
END

SYMBOLIC REFERENCE MAP (R=1)
ENTRY POINTS
6217 MAIN

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>7267</td>
<td>ALPHA</td>
<td>REAL</td>
<td>103275</td>
</tr>
<tr>
<td>103345</td>
<td>CNME2</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>103606</td>
<td>CNPE2</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>56101</td>
<td>DATA</td>
<td>COMPLEX</td>
<td>ARRAY</td>
</tr>
<tr>
<td>40477</td>
<td>DATE2</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>52415</td>
<td>DX</td>
<td>COMPLEX</td>
<td>ARRAY</td>
</tr>
<tr>
<td>42077</td>
<td>DXE2</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>SN</td>
<td>TYPE</td>
<td>RELOCATION</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>36077 DYE</td>
<td>REAL</td>
<td>ARRAY</td>
<td>45477</td>
</tr>
<tr>
<td>17555 H</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7305</td>
</tr>
<tr>
<td>7272 IEF</td>
<td>INTEGER</td>
<td>ARRAY</td>
<td>7261 IFLAG</td>
</tr>
<tr>
<td>7314 NMAX</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7271 ISF</td>
</tr>
<tr>
<td>7306 J</td>
<td>INTEGER</td>
<td></td>
<td>7310 K</td>
</tr>
<tr>
<td>100255 LINH</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7265 NFRAMES</td>
</tr>
<tr>
<td>7270 NFREQ</td>
<td>INTEGER</td>
<td></td>
<td>100655 NLINH</td>
</tr>
<tr>
<td>50077 NN</td>
<td>INTEGER</td>
<td>ARRAY</td>
<td>7307 NR</td>
</tr>
<tr>
<td>7266 NUNRS</td>
<td>INTEGER</td>
<td>57045</td>
<td>S CON</td>
</tr>
<tr>
<td>7263 NZ</td>
<td>INTEGER</td>
<td>7302 NS</td>
<td>S COMPLEX</td>
</tr>
<tr>
<td>70125 PFM</td>
<td>REAL</td>
<td>ARRAY</td>
<td>70105</td>
</tr>
<tr>
<td>70045 PHF</td>
<td>REAL</td>
<td>ARRAY</td>
<td>17351 PHIT</td>
</tr>
<tr>
<td>7451 QDROOT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>70065 QFD</td>
</tr>
<tr>
<td>7341 R</td>
<td>REAL</td>
<td>ARRAY</td>
<td>70155 RINV</td>
</tr>
<tr>
<td>17577 ROOT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7317 S</td>
</tr>
<tr>
<td>100755 SAVE</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>57045 SD</td>
</tr>
<tr>
<td>65645 SOATA</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>7274 SIGOT</td>
</tr>
<tr>
<td>7311 SUNDATA</td>
<td>REAL</td>
<td></td>
<td>7312 SUMDX</td>
</tr>
<tr>
<td>7313 SUMY</td>
<td>REAL</td>
<td>ARRAY</td>
<td>61245 SK</td>
</tr>
<tr>
<td>63435 SY</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>7277 TAF</td>
</tr>
<tr>
<td>7276 TD</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7200 TDF</td>
</tr>
<tr>
<td>7311 UC</td>
<td>REAL</td>
<td>ARRAY</td>
<td>731767 V</td>
</tr>
<tr>
<td>7301 VARAF</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7300 VARDF</td>
</tr>
<tr>
<td>7273 VARM</td>
<td>REAL</td>
<td>ARRAY</td>
<td>72757 W</td>
</tr>
<tr>
<td>52301 WORK</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>17551 WT</td>
</tr>
<tr>
<td>7176 X</td>
<td>REAL</td>
<td>ARRAY</td>
<td>70151 XFM</td>
</tr>
<tr>
<td>103155 XFM</td>
<td>REAL</td>
<td>ARRAY</td>
<td>103225 XFM2</td>
</tr>
<tr>
<td>70145 XFP</td>
<td>REAL</td>
<td>ARRAY</td>
<td>103415 XFP</td>
</tr>
<tr>
<td>103455 XFP2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7333 XMAX</td>
</tr>
<tr>
<td>7303 XSHIFT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>17341 XT</td>
</tr>
<tr>
<td>7177 Y</td>
<td>REAL</td>
<td>ARRAY</td>
<td>7336 YMAX</td>
</tr>
<tr>
<td>7304 YSHIFT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>17575 YT</td>
</tr>
</tbody>
</table>

**FILE NAMES**

- **MODE**: 0
- **INPUT**: 0
- **OUTPUT**: 2054
- **TAPE**: 4130

**EXTERNALS**

<table>
<thead>
<tr>
<th>EXTERNALS</th>
<th>TYPE</th>
<th>ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCUM</td>
<td>16</td>
<td>DERIV</td>
</tr>
<tr>
<td>FILST</td>
<td>10</td>
<td>FOURT</td>
</tr>
<tr>
<td>IDEAL</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>NOISE</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>PROP</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>SETUP</td>
<td>49</td>
<td>8</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>STATFM</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>UPDAT</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

**INLINE FUNCTIONS**

<table>
<thead>
<tr>
<th>INLINE FUNCTIONS</th>
<th>TYPE</th>
<th>ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL 2 INTRIN</td>
<td>REAL</td>
<td>1</td>
</tr>
</tbody>
</table>

**STATEMENT LABELS**

- 6377 6
- 0 7
- 0 90
- 0 106
- 0 172
SUBROUTINE SETUP

SUBROUTINE SETUP(COV, S, NZ, NNM, NRUN, ALPHA, NFREQ, 1ISF, IEF, VAR, SIGD, DT, TD, IFLAG, PHIT, QROOT, H, TAF, 2VARDF, VARAF, TDF, PHIF, QFD, DATE, DATE2, DAE, DAE2, 3DYE, DYE2, XFME, XFME2, CNME, CNME2, XFPE, XFPE2, 4CNPE, CNPE2, IMAX, IMAX(3), YMAX, YMAX(3), R(64, 64) 5REAL PHIT(8, 8), QROOT(8, 8), H(2, 8) 6REAL RROOT(64, 64), DATE(8, 8, 20), DAE(8, 8, 20), DAE2(8, 8, 20) 7REAL DYE(8, 8, 20), DYE2(8, 8, 20) 8COMPLEX SD(24, 24), SX(24, 24), SY(24, 24) 9REAL PHIF(4, 4), QFD(4, 4) 10REAL RINV(64, 64) 11REAL XFPE(2, 20), XFPE2(2, 20), CNME(2, 20), CNME2(2, 20) 12REAL CNPE(2, 20), CNPE2(2, 20)

C*
C** READ IN INITIALIZATION
C*
C********************************************************************
385 WRITE(6, 3787)
3787 FORMAT(1X, *GAUSSIAN TARGET COVARIANCE VALUE*)
READ(5, 560) CGV
560 FORMAT(F6.2)
IF(EOF(5).NE.0.) GO TO 6421
CALL INIT(COV, S, NZ, NNM, NRUN, ALPHA, NFREQ, ISF, 1IEF, VAR, SIGD, DT, TD)
30 C C 1RETS SEED FOR RANDOM # GENERATOR
CALL RANSET(12345)
35 C C DEFINE TRUTH MODEL DYNAMICS
C C C C C C
CALL TRUTH(PHIT, QROOT, H, SIGD, DT, TD)
WRITE(6, 7777) ((PHIT(I, J) J=1, 8), I=1, 8)
WRITE(6, 7777) ((QROOT(I, J) J=1, 8), I=1, 8)
7777 FORMAT(/, *(2X, 8E10.3))
C C INITIALIZE THE FILTERS PARAMETERS
C CALL INITF(TAF, VARDF, VARAF)
C C INITIALIZE THE FILTERS MATRICES DEFINITION
C C C C
CALL FILTER(TDF, VARDF, TAF, VARAF, DT, PHIF, QFD)
C C C C C C
C C C C C
C C C C C C C C
C SUBROUTINE SETUP(4.8+528) 09/21/81 12:15:51 PAGE 1
SUBROUTINE SETUP

DO 9 I=1,8
DO 9 J=1,8
DO 9 K=1,20
DATE(I,J,K)=0.
DATE2(I,J,K)=0.
DXE(I,J,K)=0.
DYE(I,J,K)=0.
DXE2(I,J,K)=0.
DYE2(I,J,K)=0.

9    INITIALIZE THE FILTER ERROR MATRICES TO ZERO

DO 21 I=1,2
DO 21 J=1,20
XFME(I,J)=0.
XFME2(I,J)=0.
CNPE(I,J)=0.
CNPE2(I,J)=0.

21   SET UP IDEAL DATA FOR ERROR CALCULATION

CALL IDEAL(IMAX,5,XMAX,YMAX,24,NZ,X,Y,SD,SX,SY)

CALL SPTN(VARM,H,B)

DO 6428 I=1,64
DO 6428 J=1,64
R(I,J)=0.0
IF((I.EQ.J)) R(I,J)=VARM
6428 CONTINUE

CALL CHOLY(R,64,RROOT)

GET THE INVERSE OF THE CORRELATION COEFFICIENT MATRIX
- NEEDED FOR THE INVERSE COV METHOD

CALL INVERT(R,64,RINV)

RETURN

IFLAG=1
RETURN
SUBROUTINE SETUP  74/74  OPT=1  PMDMP

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3  SETUP

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0  ALPHA</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  CNME</td>
</tr>
<tr>
<td>0  CNPE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  CNPE</td>
</tr>
<tr>
<td>0  CNPE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  COV</td>
</tr>
<tr>
<td>0  DATE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  DATE2</td>
</tr>
<tr>
<td>0  DT</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  DYE</td>
</tr>
<tr>
<td>0  DXE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  DYE</td>
</tr>
<tr>
<td>0  DYE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  H</td>
</tr>
<tr>
<td>330  I</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0  IEF</td>
</tr>
<tr>
<td>330  J</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0  IVAX</td>
</tr>
<tr>
<td>332  K</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0  NFRAMES</td>
</tr>
<tr>
<td>0  HNREQ</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0  NUNS</td>
</tr>
<tr>
<td>0  NZM</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0  NZM</td>
</tr>
<tr>
<td>0  PHIT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  PHIT</td>
</tr>
<tr>
<td>0  QD</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  QD</td>
</tr>
<tr>
<td>0  RNV3</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  RINV</td>
</tr>
<tr>
<td>0  RNV4</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  RNV</td>
</tr>
<tr>
<td>0  SMD</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  SMD</td>
</tr>
<tr>
<td>0  S5D</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>0  S5D</td>
</tr>
<tr>
<td>0  S5X</td>
<td>COMPLEX</td>
<td>ARRAY</td>
<td>0  S5X</td>
</tr>
<tr>
<td>0  SDF</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  TO</td>
</tr>
<tr>
<td>0  TDF</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  VARF</td>
</tr>
<tr>
<td>0  VARDF</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  VARM</td>
</tr>
<tr>
<td>0  X</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  XFPE</td>
</tr>
<tr>
<td>0  XFPE2</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0  XFE</td>
</tr>
<tr>
<td>0  Y</td>
<td>REAL</td>
<td>F.P.</td>
<td>0  YXAX</td>
</tr>
</tbody>
</table>

FILE NAMES

<table>
<thead>
<tr>
<th>TAPES</th>
<th>MODE</th>
<th>TAPE6</th>
<th>FMT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXTERNALS

<table>
<thead>
<tr>
<th>CHOLY</th>
<th>TYPE</th>
<th>ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>FILTER</td>
<td>TYPE</td>
<td>IDEAL</td>
</tr>
<tr>
<td>INIT</td>
<td>TYPE</td>
<td>INITF</td>
</tr>
<tr>
<td>INVERT</td>
<td>TYPE</td>
<td>RANSET</td>
</tr>
<tr>
<td>SPIN</td>
<td>TYPE</td>
<td>TRUTH</td>
</tr>
</tbody>
</table>

STATEMENT LABELS

| 0 9  | 3787 | FMT | 0 | 21 | 316 | 560 | FMT |
| 304  | 3985 | INACTIVE | 201 | 6421 |   |
| 0 6428 | INACTIVE | 320 | 7777 | FMT | NO REFS |

LOOPS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>0</td>
<td>1</td>
<td>60</td>
<td>68</td>
</tr>
<tr>
<td>103</td>
<td>0</td>
<td>J</td>
<td>61</td>
<td>68</td>
</tr>
</tbody>
</table>
SUBROUTINE ACCUM( 74/74 OPT=1 PMOMP 

1 SUBROUTINE ACCUM(SUMDATA,SAVE,SD,SUMDX,DX,SX,SUMDY,DY,SY,
#DATE,DATE2,DXE,DEXE,DYE,DYE2,NR).
COMPLEX SAVE(24,24),SD(24,24),SX(24,24),SY(24,24)
COMPLEX DX(24,24),DY(24,24)
5 REAL DATE(8,8,20),DATE2(8,8,20),DXE(8,8,20),DEXE(8,8,20)
REAL DYE(8,8,20),DYE2(8,8,20)
ACCUMULATE SUM AND SUM OF SQUARE ERRORS IN INTENSITY AND DERIVATIVES
FOR EACH OF THE CENTER 8x8 PIXELS

10 DO 2 I=9,16
DO 2 J=9,16
15 SUMDATA=
1(ABS(REAL(SAVE(I,J)))-ABS(REAL(SD(I,J))))
SUX=
#ABS(REAL(DX(I,J)))-ABS(REAL(SX(I,J))))
SUXD=
20 DATE(I-8,J-8,NR)=DATE(1-8,J-8,NR)+SUMDATA
DATE2(I-8,J-8,NR)=DATE2(1-8,J-8,NR)+SUMDATA**2
DXE(I-8,J-8,NR)=DXE(1-8,J-8,NR)+SUXDX
DEXE2(I-8,J-8,NR)=DEXE2(1-8,J-8,NR)+SUXD**2
DYE(I-8,J-8,NR)=DYE(1-8,J-8,NR)+SUMDY
DYE2(I-8,J-8,NR)=DYE2(1-8,J-8,NR)+SUMD**2
25 CONTINUE
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 ACCUM

VARIABLES SN TYPE RELOCATION
0 DATE REAL ARRAY F.P. 0 DATE2 REAL ARRAY F.P.
0 DX COMPLEX ARRAY F.P. 0 DXE REAL ARRAY F.P.
0 DXE2 REAL ARRAY F.P. 0 DYE COMPLEX ARRAY F.P.
0 DYE REAL ARRAY F.P. 0 DYE2 REAL ARRAY F.P.
50 i INTEGER ARRAY F.P. 51 j INTEGER ARRAY F.P.
0 NR INTEGER ARRAY F.P. 0 SAVE COMPLEX ARRAY F.P.
0 SD COMPLEX ARRAY F.P. 0 SUMDATA REAL ARRAY F.P.
0 SUMDX REAL ARRAY F.P. 0 SUMDY REAL ARRAY F.P.
0 SX COMPLEX ARRAY F.P. 0 SY COMPLEX ARRAY F.P.

INLINE FUNCTIONS TYPE ARGS
ABS REAL 1 INTRIN

STATEMENT LABELS
0 2
SUBROUTINE PERF(X,Y,XSHIFT,YSHIFT,SD,SX,SY,IMAX,XMAX,YMAX,S,NZ)
COMPLEX SD(24,24),SX(24,24),SY(24,24)
REAL IMAX(3),XMAX(3),YMAX(3),S(12)

5 THIS ROUTINE COMPUTES THE POEFECT DATA WITHIN A 24X24 ARRAY SD
 WHICH IS OFFSET FROM THE CENTER OF THE FOV BY XSHIFT AND
 YSHIFT WITHOUT ANY TRANSFORMS IT ALSO COMPUTES THE DERIVATIVE

10 XMAX(1)=X+4.*XSHIFT
XMAX(2)=X+2.*XSHIFT
XMAX(3)=X+6.*XSHIFT
YMAX(1)=Y+4.*YSHIFT-2.6666666
YMAX(2)=Y+4.*YSHIFT+1.333333
YMAX(3)=Y+4.*YSHIFT+1.333333

15 CALL IDEAL(IMAX,S,XMAX,YMAX,24,NZ,X,Y,SD,SX,SY)
DO 1 I=1,24
DO 1 J=1,24
SX(I,J)=SX(I,J)
1 SY(I,J)=SY(I,J)

20 RETURN
END

SYMBOLIC REFERENCE MAP (R=1)
ENTRY POINTS
3 PERF

VARIABLES SN TYPE RELOCATION
107 I INTEGER 0 IMAX REAL ARRAY F.P.
110 J INTEGER 0 NZ INTEGER ARRAY F.P.
0 S REAL 0 SD COMPLEX ARRAY F.P.
0 SX COMPLEX 0 SY COMPLEX ARRAY F.P.
0 X REAL 0 XMAX REAL ARRAY F.P.
0 XSHIFT REAL 0 Y REAL F.P.
0 YMAX REAL 0 YSHIFT REAL F.P.

EXTERNALS TYPL ARGS
IDEAL 11

STATEMENT LABELS
0 1

LOOPS LABEL INDEX FROM-TO LENGTH PROPERTIES
43 1 I 16 19 209 NOT INNER
54 1 J 17 19 58 INSTACK

STATISTICS
PROGRAM LENGTH 1158 77
520008 CM USED
SUBROUTINE STATFM 74/74 OPT=1 PMDP 

1 SUBROUTINE STATFM(XFME, XFME2, CNME, CNME2, XFMT, XT, YT, NR)
REAL XFME(2,20), XFME2(2,20), CNME(2,20), CNME2(2,20)
REAL XFMT(4), XT(8,1), YT(2,1)

5 THIS ROUTINE GATHERS THE INFORMATION THAT WILL BE
REQUIRED TO COMPUTE THE STATISTICS OF THE PREDICTIONS
OF THE FILTER PRIOR TO MEASUREMENT INCORPORATION

10 FIRST COLLECT THE ERROR IN THE PREDICTED DYNAMIC LOCATION

15 XFME1(NR)=XFME1(NR)+XFME(1)*XT(1,1)
XFME2(NR)=XFME2(NR)+XFME(2)*XT(2,1)

20 COLLECT ERROR IN CENTROID PREDICTED LOCATION MINUS

CNME1(NR)=CNME1(NR)+(XFMT(1)+XFMT(3)-YT(1,1))
CNME2(NR)=CNME2(NR)+(XFMT(2)+XFMT(4)-YT(2,1))

25 COLLECT THE SQUARE OF THE ERROR

CNME1(NR)=CNME1(NR)+(XFMT(1)+XFMT(3)-YT(1,1))^2
CNME2(NR)=CNME2(NR)+(XFMT(2)+XFMT(4)-YT(2,1))^2
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 STATFM

VARIABLES SN TYPE RELOCATION
0 CVME REAL ARRAY F.P.
0 NR INTEGER ARRAY F.P.
0 XFME REAL ARRAY F.P.
0 XT REAL ARRAY F.P.
0 CNME REAL ARRAY F.P.
0 XFME2 REAL ARRAY F.P.
0 YT REAL ARRAY F.P.

STATISTICS
PROGRAM LENGTH 438 35
52000B CM USED
SUBROUTINE STATFP

REAL XFPE(2,20), XFPE2(2,20), CNPE(2,20), CNPE2(2,20)
REAL XT(8,1), YT(2,1), XFP(4)

THIS ROUTINE GATHERS THE INFORMATION THAT WILL BE
REQUIRED TO COMPUTE THE STATISTICS ON THE FILTERS
UPDATED STATE ESTIMATES

COMPUTE DIFFERENCES THAT WILL BE NEEDED

DIF1=XT(1,1) - XT(1,1)
DIF2=XT(2,1) - XT(2,1)
DIF3=XT(1,1) + XT(1,1)
DIF4=XT(2,1) + XT(2,1)

FIRST COLLECT THE ERROR IN THE DYNAMIC LOCATION ESTIMATES

XFPE(1, NR) = XFPE(1, NR) + DIF1
XFPE(2, NR) = XFPE(2, NR) + DIF2

NOW COLLECT THE SQUARE OF THAT ERROR

XFPE2(1, NR) = XFPE2(1, NR) + (DIF1)**2
XFPE2(2, NR) = XFPE2(2, NR) + (DIF2)**2

NOW COLLECT THE ERROR IN THE CENTROID UPDATE

CNPE(1, NR) = CNPE(1, NR) + DIF3
CNPE(2, NR) = CNPE(2, NR) + DIF4

NOW COLLECT THE ERROR SQUARED

CNPE2(1, NR) = CNPE2(1, NR) + (DIF3)**2
CNPE2(2, NR) = CNPE2(2, NR) + (DIF4)**2

RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

<table>
<thead>
<tr>
<th>ENTRY POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 STATFP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>ARRAY</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CNPE</td>
<td>REAL</td>
<td>ARRAY</td>
<td>F.P.</td>
<td>0 CNPE2</td>
</tr>
<tr>
<td>46 DIF1</td>
<td>REAL</td>
<td></td>
<td></td>
<td>45 DIF2</td>
</tr>
<tr>
<td>46 DIF3</td>
<td>REAL</td>
<td></td>
<td></td>
<td>47 DIF4</td>
</tr>
<tr>
<td>0 NR</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0 XFP</td>
<td>REAL</td>
</tr>
<tr>
<td>0 XFPE</td>
<td>REAL</td>
<td>ARRAY</td>
<td>F.P.</td>
<td>0 XFPE2</td>
</tr>
<tr>
<td>0 XT</td>
<td>REAL</td>
<td>ARRAY</td>
<td>F.P.</td>
<td>0 YT</td>
</tr>
</tbody>
</table>
SUBROUTINE FILST

DO 1 I=1,2
DO 1 J=1,NFRAMES
XFME(1,I)*XFME(1,J)/FLOAT(NRUNS)
XFPE(1,I)*XFPE(1,J)/FLOAT(NRUNS)
CNME(1,I)*CNME(1,J)/FLOAT(NRUNS)
CNPE(1,I)*CNPE(1,J)/FLOAT(NRUNS)
XFME2(1,J)=SORT(ABS(XFME2(1,J)/FLOAT(NRUNS)-XFME(1,J)**2))
XFPE2(1,J)=SORT(ABS(XFPE2(1,J)/FLOAT(NRUNS)-XFPE(1,J)**2))
CNME2(1,J)=SORT(ABS(CNME2(1,J)/FLOAT(NRUNS)-CNME(1,J)**2))
CNPE2(1,J)=SORT(ABS(CNPE2(1,J)/FLOAT(NRUNS)-CNPE(1,J)**2))
WRITE(6,2)
2 FORMAT(T2,FRAME*,T10,*XERR(-)*,T30,*SXERR(-)*,T55,*YERR(-)*, *
*UB*,*SYERR(-)*)
DO 3 I=1,NFRAMES
WRITE(6,4) I,XFME(1,I),XFME2(1,I),XFME(2,I)
WRITE(6,2)
4 FORMAT(T4,12,T9,E12.5,T28,E12.5,T52,E12.5,T66,E12.5)
CONTINUE
5 FORMAT(T2,FRAME*,T10,*XERR(+)*,T30,*SXERR(+)*,T55,*YERR(+)*, *
*UB*,*SYERR(+)*)
DO 6 I=1,NFRAMES
WRITE(6,7) I,XFPE(1,I),XFPE2(1,I),XFPE(2,I),XFPE(2,I)
WRITE(6,2)
7 FORMAT(T4,12,T9,E12.5,T28,E12.5,T52,E12.5,T66,E12.5)
CONTINUE
WRITE(6,101)
101 FORMAT(T2,FRAME*,T10,*CNER(-)*,T30,*SCNER(-)*,T55,*YCER(-)*, *
*UB*,*YCER(-)*)
DO 102 I=1,NFRAMES
WRITE(6,103) I,CNME(1,I),CNME2(1,I),CNME(2,I),CNME2(2,I)
WRITE(6,2)
103 FORMAT(T4,12,T9,E12.5,T28,E12.5,T52,E12.5,T66,E12.5)
CONTINUE
WRITE(6,10)
10 FORMAT(T2,FRAME*,T10,*CNER(+)*,T30,*SCNER(+)*,T55,*YCER(+)*, *
*UB*,*YCER(+)*)
DO 103 I=1,NFRAMES
WRITE(6,9) I,CNPE(1,I),CNPE2(1,I),CNPE(2,I),CNPE2(2,I)
WRITE(6,2)
9 FORMAT(T4,12,T9,E12.5,T28,E12.5,T52,E12.5,T66,E12.5)
CONTINUE
RETURN
END
SUBROUTINE FILST  74/74  OPT=1  PMDMP  FTN 4.8+52B  09/21/81  12:15:51  PAGE 2

ENTRY POINTS
3 FILST

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CNAME</td>
<td></td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>0 CNPE</td>
<td></td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>317 I</td>
<td></td>
<td>INTEGER</td>
<td>320 J INTEGER</td>
</tr>
<tr>
<td>0 NFRAMES</td>
<td></td>
<td>INTEGER</td>
<td>0 NRUNS INTEGER</td>
</tr>
<tr>
<td>0 XFME</td>
<td></td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>0 XFPE</td>
<td></td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
</tbody>
</table>

FILE NAMES
TAPE6 MODE FMT

EXTERNALS
SORT TYPE ARGS
REAL 1 LIBRARY

INLINE FUNCTIONS
ABS TYPE ARGS
REAL 1 INTRIN

STATEMENT LABELS
| 0 1 | 152 2 | FMT | 0 3 |
| 173 4 | 234 5 | FMT | 0 6 |
| 225 7 | 0 8 | FMT | 311 9 | FMT |
| 270 10 | 236 101 | FMT | 0 102 |
| 257 103 | FMT |

LOOPS
<table>
<thead>
<tr>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 J</td>
<td>I</td>
<td>8 17</td>
<td>479</td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>10 L</td>
<td>I</td>
<td>9 17</td>
<td>446</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>61 J</td>
<td>I</td>
<td>21 25</td>
<td>128</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>77 J</td>
<td>I</td>
<td>29 32</td>
<td>148</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>115 L</td>
<td>I</td>
<td>36 39</td>
<td>148</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>133 R</td>
<td>I</td>
<td>43 46</td>
<td>148</td>
<td>EXT REFS</td>
</tr>
</tbody>
</table>

STATISTICS
PROGRAM LENGTH 3738 251
52000B CM USED
SUBROUTINE INITF(TAF,VARDF,VARAF)

THIS ROUTINE CONTROLS INPUTING VALUES NEEDED FOR THE KALMAN FILTER

TAF=.07072

THIS IS THE CORRELATION TIME FOR THE ATMOSPHERIC MODEL FOR THE FILTER

VARDF=1.0

THE VARIANCE OF THE ATMOSPHERIC JITTER FOR THE FILTER

VARAF=0.001

RETURN

END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 INITF

VARIABLES SN TYPE RELOCATION
0 TAF REAL F.P.
0 VARDF REAL F.P.

STATISTICS
PROGRAM LENGTH 15B 13
52000B CM USED
SUBROUTINE FILTER

1 SUBROUTINE FILTER(TDF,VARDF,TAF,VARAF,DT,PHIF,QFD)
REAL PHIF(4,4),QFD(4,4)

C THIS ROUTINE SETS UP THE STATE TRANSITION MATRIX AND QFD MATRIX

5 TDF CORRELATION TIME FOR THE TARGET DYNAMICS
C VARDF TARGET DYNAMICS NOISE VARIANCE
C VARAF ATMOSPHERIC NOISE VARIANCE

10 D(XF(T))/DT-TIME DERIVATIVE OF FILTER STATE
C FILFT-FILTER PLANT MATRIX
C WFT INPUT WHITE NOISE VECTOR FOR FILTER
C STATE SPACE MODEL

15 D(XF(T))/DT=FILFT*XF(T)+WFT(T)

WHERE

20 FILFT =
: : 0 0 -1/TDF 0 0 :
: : 0 0 0 -1/TAF 0 :
: : 0 0 0 0 -1/TAF :

E\*WFT(T)=0
E\*WFT(T)!=QF

WHERE

25 QF =
: : 0 0 2*VARDF/TDF 0 0 :
: : 0 2*VARDF/TDF 0 0 :
: : 0 0 2*VARAF/TAF 0 0 :
: : 0 0 0 0 2*VARAF/TAF :

THE SOLUTION TO THE DYNAMIC EQUATIONS

35 XF(I+1)=PHIF*XF(I)

FOR PROPAGATION OF COVARIANCE MATRIX NEED QFD

DO 1 I=1,4
DO 1 J=1,4
PHIF(I,J)=0.
1 QFD(I,J)=0.
PHIF(1,1)=EXP(-DT/TDF)
PHIF(2,2)=PHIF(1,1)
PHIF(3,3)=EXP(-DT/TDF)
PHIF(4,4)=PHIF(3,3)
QFD(1,1)=VARDF(1.-EXP(-2.*DT/TDF))
QFD(2,2)=QFD(1,1)
QFD(3,3)=VARAF(1.-EXP(-2.*DT/TAF))
QFD(4,4)=QFD(3,3)
RETURN
END
SUBROUTINE FILTER 74/74 OPT=1 PMDMP FTN 4.8+52B 09/21/81 12.15.51 PAGE 2

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 FILTER

VARIABLES  SN  TYPE  RELOCATION
0  DT  REAL  F.P.
53  J  INTEGER
0  QFD  REAL  ARRAY  F.P.
0  TDF  REAL  F.P.
0  VARDF  REAL  F.P.

EXTERNALS  TYPE  ARGS
EXP  REAL  1 LIBRARY

STATEMENT LABELS
0  1

LOOPS  LABEL  INDEX  FROM-TO  LENGTH  PROPERTIES
7  1  I  39 42  148  NOT INNER
15  1  J  40 42  38  INSTACK

STATISTICS
PROGRAM LENGTH  60B  48
52000B CM USED
SUBROUTINE PROP

1  SUBROUTINE PROP(PHI, QFD, PFP, PFM, XFP, XF)
   REAL PHI(4,4), QFD(4,4), PFP(4,4), PFM(4,4), XFP(4), XF(4)
   REAL TEMP(4,4), TEMP2(4,4)

5  THIS ROUTINE IMPLEMENTS THE STATE TRANSITION
   - EQUATIONS FOR THE FILTER
   X F(I+1) = PHI * X F(I)
   PI = PHI * PFP + PHI * QFD

10  WHERE
   PHI = FILTER STATE TRANSITION MATRIX
   X F = FILTER STATE VECTOR
   PFM = COV FILTER STATES MINUS
   PFP = COV FILTER STATES PLUS

15  WHERE
   QFD = VARDF * (1. - EXP(-2.*DT/TDF)) 0 0 0 0 ;
    : 0 0 VARDF * (1. - EXP(-2.*DT/TDF)) 0 0 0 ;
    : 0 0 0 VARAF * (1. - EXP(-2.*DT/TDF)) 0 ;
    : 0 0 0 0 VARAF * (1. - EXP(-2.*)) ;

20  PERFORM FILTER STATE PROPAGATION

25  CALL MULT(PHI, XFP, 4, 4, 1, XF)
   ADD DETERMINISTIC INPUT OF 3 PIXELS PER TIME HISTORY IN THE X DIR
   XF(I+1) = XF(I) + 15
   CALL MULT(PHI, PFP, 4, 4, 4, TEMP1)
   CALL MULT(TEMP1, PHI, 4, 4, 4, TEMP2)
   DO 1 I = 1, 4
   DO 1 J = 1, 4
   1 PFM(I, J) = TEMP2(I, J) + QFD(I, J)
   RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 PROP

VARIABLES SN TYPE RELLOCATION
75 I INTEGER 75 J INTEGER
0 PFM REAL ARRAY F.P. 0 PFP REAL ARRAY F.P.
0 PHI REAL ARRAY F.P. 0 QFD REAL ARRAY F.P.
77 TEMP REAL ARRAY F.P. 117 TEMP2 REAL ARRAY F.P.
0 XF REAL ARRAY F.P. 0 XFP REAL ARRAY F.P.

EXTERNALS TYPE ARGS
MULT 6

STATEMENT LABELS
0 1
SUBROUTINE UPD(2,LINH,NLINH,XFP,XFM,PFP,PFM,RINV)
REAL Z(64),LINH(64),NLINH(64),LINHT(64,64),XFP(4),XFM(4)
REAL PFP(4,4),PFM(4,4),RINV(64,64),TEMP1(64,64),TEMP2(4,4)
REAL RESID(64),PINV(4,4),GAINK(4,64),UPD(4)

C THIS ROUTINE PROCESSES ONE MEASUREMENT VECTOR Z
C AND ALSO UPDATES THE FILTER STATES XFP
C Z IS THE MEASUREMENT VECTOR OF FLIR MEASUREMENTS
C LINH IS THE NONLINEAR INTENSITY FUNCTION AFTER SMOOTHING
C LINHT IS THE LINEAR INTENSITY FUNCTION
C XFP IS FILTER STATES PLUS
C XFM IS FILTER STATES MINUS
C PFP COV MATRIX OF FILTER STATES PLUS
C PFM COV MATRIX OF FILTER STATES MINUS
C RINV INVERSE OF SPATIAL CORRELATION COEF MATRIX
C TEMP1 WILL HOLD H-TRANSPOSE*RINV
C PINV WILL HOLD COV MATRIX MINUS INVERSED
C TEMP2 WILL HOLD M*RINV
C GAINK KALMAN FILTER GAIN
C RESID RESIDUAL
C UPD GAIN TIMES RESIDUAL
C
C THE EQUATIONS USED FOR THE UPDATE ARE
C FIRST FOR COMPUTING THE COVARIANCE PLUS MATRIX
C USING THE INVERSE COV METHOD
C
C PINV(1+)=PINV(1-)+LINHT(1)*RINV(1)LINH(1)
PFP(1+)=PFP(1-)+1
K(I+)=P(I+)*LINHT(1)*RINV(I)
XFP(I+)=XFM(I-)+K(I+)*Z(I)-NLIH(I)

C FIRST CREATE LINH TRANSPOSED
C
DO 1 I=1,64
DO 1 J=1,4
LINH*(I,J)=LINH(I,J)
1
C COMPUTE THE INVERSE OF THE COV PLUS MATRIX
CALL MULT(LINHT,RINV,64,64,TEMP1)
CALL MULT(TEMP1,LINH,64,64,TEMP2)

C INVERT THE COVARIANCE MINUS MATRIX
CALL INVERT(PFM,4,PINV)
C
DO 2 I=1,4
DO 2 J=1,4
PINV(I,J)=PINV(I,J)+TEMP2(I,J)
2
C CREATE P(I+)=PFP
CALL INVERT(PFM,4,PFP)
C
C GAIN TIMES FILTER GAIN
C
**JUBOUTINE UPDAT**  74/74   OPT=1   PMOMP

<table>
<thead>
<tr>
<th>C</th>
<th>COMPUTE STATE MEASUREMENT UPDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>DO 3 I=1,64</td>
</tr>
<tr>
<td>3</td>
<td>RESID(I)=Z(I)-NLINH(I)</td>
</tr>
<tr>
<td></td>
<td>CALL MULT(GAINK,RESID,4,64,1,UPD)</td>
</tr>
<tr>
<td>65</td>
<td>DO 4 I=1,4</td>
</tr>
<tr>
<td>4</td>
<td>XFP(I)*XFM(I)+UPD(I)</td>
</tr>
<tr>
<td></td>
<td>RETURN</td>
</tr>
<tr>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>

**SYMBOLIC REFERENCE MAP (R=1)**

**ENTRY POINTS**

**3 UPDAT**

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAINK</td>
<td>1312</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>J</td>
<td>151</td>
<td>INTEGER</td>
<td>0 LINH REAL</td>
</tr>
<tr>
<td>LINH</td>
<td>152</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>PFM</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>PINV</td>
<td>1272</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>PINV</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>TEMP2</td>
<td>1152</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>XFM</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXTERNALS</th>
<th>TYPE</th>
<th>ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVERT</td>
<td>REAL</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATEMENT LABELS</th>
<th>0</th>
<th>2</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>I</td>
<td>37 39</td>
<td>14B</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>15</td>
<td>I</td>
<td>38 39</td>
<td>38</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>I</td>
<td>49 51</td>
<td>14B</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>J</td>
<td>50 51</td>
<td>3B</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>3</td>
<td>60 61</td>
<td>3B</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>4</td>
<td>63 64</td>
<td>39</td>
<td>INSTACK</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATISTICS</th>
<th>PROGRAM LENGTH</th>
<th>1720B</th>
<th>976</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>520008 CM USED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE STAT(NFRAMES, DATE, NRUNS, DYE, DXE, DATE2, DXE2, DYE2, NZ, 
NFRQ, COV, VARM, ALPHA, XSHIFT, YSHIFT, SIGDT)
REAL DATE(8, 8, 20), DXE(8, 8, 20), DYE(8, 8, 20), DATE2(8, 8, 20),
DYE2(8, 8, 20), DXE2(8, 8, 20)
DO 10 K = 1, NFRAMES
DO 10 I = 1, 8
DO 10 J = 1, 8
DATE(I, J, K) = DATE(I, J, K) / NRUNS
DXE(I, J, K) = DXE(I, J, K) / NRUNS
DYE(I, J, K) = DYE(I, J, K) / NRUNS
DATE2(I, J, K) = DATE2(I, J, K) / NRUNS
DYE2(I, J, K) = DYE2(I, J, K) / NRUNS
CONTINUE
WRITE(6, 9987) NRUNS, NFRAMES, NZ, NFRQ, COV, VARM, ALPHA, 
SIGDT 9987 FORMAT(1H1, 110, *NRUNS**, 12, T40, *FRAMES**, 12, T70, *NUMBER ZERO PAO**, 
# 11, T100, *NUMBER FREQ ZERODE**, 12, /, T10, *GAUSSIAN COVARIANCE**, 
# F5.2, T40, *BACKGROUND VARIANCE**, F5.1, T70, *SMOOTHING ALPHA**, 
F7.3, /,
# T70, *TRUTH MODEL UNCERTAINTY**, F5.2, ///)
DO 11 K = 1, NFRAMES
C WRITE(6, 9988) ((DATE(I, J, K), J = 1, 8), I = 1, 8)
C WRITE(6, 9988) ((DXE(I, J, K), J = 1, 8), I = 1, 8)
C WRITE(6, 9988) ((DYE(I, J, K), J = 1, 8), I = 1, 8)
C WRITE(5, 9988) ((DATE2(I, J, K), J = 1, 8), I = 1, 8)
C WRITE(5, 9988) ((DXE2(I, J, K), J = 1, 8), I = 1, 8)
C WRITE(6, 9988) ((DYE2(I, J, K), J = 1, 8), I = 1, 8)
CONTINUE
9988 FORMAT((/10I8), (21, 8E16.5))
WRITE(6, 9324)
9324 FORMAT(T2, *FRAME*, T15, *MEAN ERROR*, T33, *MEAN VARIANCE*, T54, 
# *MEAN ERROR*, T71, *MEAN VARIANCE OF*, T94, *MEAN ERROR*, T111, 
# *MEAN VARIANCE OF*, T32, *OFF ERROR IN H*, T56, *IN D/DX*, 
DO 13 K = 1, NFRAMES
SUNH = 0.
VARN = 0.
SUNDX = 0.
SUNDY = 0.
VARDX = 0.
VARDY = 0.
DO 12 1 = 1, 8
DO 12 J = 1, 8
SUNH = SUNH + DATE(I, J, K) / 64.
VARN = VARN + DATE2(I, J, K) / 64.
SUNDX = SUNDX + DXE(I, J, K) / 64.
VARDX = VARDX + DYE2(I, J, K) / 64.
SUNDY = SUNDY + DYE(I, J, K) / 64.
VARDY = VARDY + DYE2(I, J, K) / 64.
CONTINUE
WRITE(6, 9325) K, SUNH, VARN, SUNDX, SUNDY, VARDX, VARDY
9325 FORMAT(T4, 12, T13, T12.5, T33, T12.5, T52, T12.5, T73, T12.5, T93, 
# T12.5, T113, T12.5)
CONTINUE
RETURN
END
### SYMBOLIC REFERENCE MAP (R=1)

#### ENTRY POINTS

<table>
<thead>
<tr>
<th>3 STAT</th>
</tr>
</thead>
</table>

#### VARIABLES SN TYPE RELOCATION

| 0 ALPHA REAL F.P. 0 COV REAL F.P. |
| 0 DATE REAL ARRAY F.P. 0 DATE2 REAL ARRAY F.P. |
| 0 DEX REAL ARRAY F.P. 0 DEX2 REAL ARRAY F.P. |
| 0 DYE REAL ARRAY F.P. 0 DYE2 REAL ARRAY F.P. |
| 247 1 INTEGER 250 J INTEGER |
| 246 K INTEGER 0 NFAMES INTEGER F.P. |
| 0 NFIQ INTEGER F.P. 0 NUNGS INTEGER F.P. |
| 0 NZ INTEGER F.P. 0 SIGDT REAL F.P. |
| 253 SMDX REAL 255 SMDY REAL |
| 251 SUMH REAL 254 VARIX REAL |
| 256 VARY REAL 252 VARH REAL |
| 0 VARH REAL F.P. 0 XSHIFT REAL *UNUSED F.P. |
| 0 YSHIF REAL *UNUSED F.P. |

#### FILE NAMES MODE

| TAPIE |

#### STATEMENT LABELS

| 0 10 0 11 0 12 |
| 0 13 171 9324 FMT 234 9325 FMT |
| 132 9987 FMT 163 9988 FMT NO REFS |

#### LOOPS LABEL INDEX FROM-TO LENGTH PROPERTIES

| 7 10 K 5 14 328 NOT INNER |
| 10 10 I 6 14 278 NOT INNER |
| 21 10 J 7 14 128 DPT |
| 16 11 K 22 29 18 INSTACK |
| 53 13 K 36 55 438 EXT REFS NOT INNER |
| 60 12 I 43 51 318 NOT INNER |
| 67 12 J 44 51 178 DPT |

#### STATISTICS PROGRAM LENGTH

| 3008 192 |

520008 CM USED
SUBROUTINE INIT

SUBROUTINE INIT(COV, S, NZ, NZM, NFRAMES, NRUNS, ALPHA, NFREQ, ISF,
# IEF, VARM, SIGDT, DT, TO)
DIMENSION S(12)

C
C DEFINE TRUE TARGET AS 3 INDEP GAUSS FUNCTIONS WITH VAR=COV
C
10 S(1)=1./COV
S(4)=S(1)
S(5)=S(1)
S(8)=S(1)
S(9)=S(1)
S(12)=S(1)
WRITE(6,3791)
3791 FORMAT(1X,'NUMBER OF ZEROS TO PAD*)
READ(5,561) NZ
561 FORMAT(12)
20 NZM=25-NZ
WRITE(6,3792)
3792 FORMAT(1X,'NUMBER OF FRAMES*)
READ(5,561) NFRAMES
WRITE(6,4023)

25 FORMAT(1X,'NUMBER OF SIMULATIONS*)
READ(5,561) NRUNS
WRITE(6,3793)
3793 FORMAT(1X,'ALPHA FOR SMOOTHING*)
READ(5,552) ALPHA
WRITE(6,3795)
3795 FORMAT(1X,'NUMBER OF HIGH FREQ COMPONENTS TO ZERO*)
READ(5,561) NFREQ
ISF=14-NFREQ
JE=12-NFREQ
WRITE(6,3789)
3789 FORMAT(1X,'INPUT BACKGROUND VARIANCE*)
READ(5,560) VARM

560 FORMAT(16.2)
WRITE(6,965)
965 FORMAT(2X,'VARIANCE OF TRUTH MODEL DYNAMICS UNCERTAINTY*)
READ(5,410) SIGDT
410 FORMAT(16.2)
DT=.0333333
TO=1.

45 RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 INIT
<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>0</td>
<td>REAL</td>
<td>F.P.</td>
<td>0</td>
<td>COV</td>
<td>REAL</td>
</tr>
<tr>
<td>DT</td>
<td>0</td>
<td>REAL</td>
<td>F.P.</td>
<td>0</td>
<td>IEF</td>
<td>INTEGER</td>
</tr>
<tr>
<td>ISF</td>
<td>0</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0</td>
<td>NFRAMES</td>
<td>INTEGER</td>
</tr>
<tr>
<td>NFREQ</td>
<td>0</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0</td>
<td>NRUNS</td>
<td>INTEGER</td>
</tr>
<tr>
<td>NZ</td>
<td>0</td>
<td>INTEGER</td>
<td>F.P.</td>
<td>0</td>
<td>NZM</td>
<td>INTEGER</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY</td>
<td>0</td>
<td>SIGOT</td>
<td>REAL</td>
</tr>
<tr>
<td>TD</td>
<td>0</td>
<td>REAL</td>
<td>F.P.</td>
<td>0</td>
<td>VARM</td>
<td>REAL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FILE NAMES</th>
<th>MODE</th>
<th>TAPE 5</th>
<th>TAPE 6</th>
<th>FMT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATEMENT LABELS</th>
<th>MODE</th>
<th>TAPE 5</th>
<th>TAPE 6</th>
<th>FMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>217 410</td>
<td>FMT</td>
<td>176</td>
<td>560</td>
<td>FMT</td>
</tr>
<tr>
<td>203 965</td>
<td>FMT</td>
<td>164</td>
<td>3789</td>
<td>FMT</td>
</tr>
<tr>
<td>102 3792</td>
<td>FMT</td>
<td>132</td>
<td>3793</td>
<td>FMT</td>
</tr>
<tr>
<td>146 3795</td>
<td>FMT</td>
<td>116</td>
<td>4023</td>
<td>FMT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATISTICS</th>
<th>LENGTH</th>
<th>CM USED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2238</td>
<td>147</td>
</tr>
</tbody>
</table>

52000B
SUBROUTINE TRUTH

SUBROUTINE TRUTH(PHIT, QDROOT, H, SIGD, DT, TD)
REAL PHIT(8,8), QD(8,8), K, QDROOT(8,8), H(2,8)
REAL QDPAR(6,6)
REAL QDPRT(6,6)

ASSUME TARGET DYNAMICS ARE:
XD = WD/(S+1/TD)
YD = WD/(S+1/TD)
XA = W+A*+B**(2/(S+A))/(S+B)**2
YA = XA

THEN AN EQUIVALENT STATE SPACE MODEL IS:
D(XD)/DT = FT*XD + GT*WT AND YD = H*XD

WHERE

FT =

GT =

HT =

XD =

THE SOLUTION OF THE DYNAMIC EQUATIONS IS

XD(I+1) = PHIT*XD(I) + SIGD(QD)*WT

WHERE PHIT = STATE TRANSITION MATRIX
SIGD = DYNAMIC NOISE STANDARD DEVIATION
WT = GAUSSIAN NOISE VECTOR
C
C NSTATE=8
K=.382109544+SIGDT
A=14.14
B=.659.5
65 DO J=1,NSTATE
DO I=1,NSTATE
PHIT(I,J)=0.
1 QUIT(I,J)=0.
PHIT(1,1)=EXP(-DT/D0)
PHIT(2,2)=EXP(-DT/D0)
70 C THESE NEXT TWO CARDS CHANGE THE TRUTH MODEL TO A DETERMINISTIC
C CONSTANT VELOCITY MODEL OF 3 PIXELS PER TIME HISTORY IN THE X D
PHIT(1,1)=1.
PHIT(2,2)=PHIT(1,1)
PHIT(3,3)=EXP(-A0.DT)
PHIT(4,4)=EXP(-B0.DT)
PHIT(5,5)=DT.EXP(0.0.DT)
PHIT(6,6)=EXP(-A0.DT)
PHIT(7,7)=EXP(-B0.DT)
PHIT(8,8)=EXP(-B0.DT)
FACT=(K+2)*(A+2)*(B+2)
FACT1=A-B
FACT2=A+B
FACT3=2.*B
G1=FACT/FACT1**4)
G2=FACT/FACT1**3)
G3=FACT/FACT1**2)
P1=1.-EXP(-2.0.A0.DT)
P2=1.-EXP(-FACT20.DT)
P3=1.-EXP(-2.0.B0.DT)
P4=DT.EXP(-FACT20.DT)
P5=DT.EXP(-2.0.B0.DT)
95 C QD(1,1)=(SIGDT)*2.+((1.-EXP(-2.0.DT/T)))*DT/2.
C BY SETTING THESE ELEMENTS OF QD = 0 ONLY A PERFECT CONSTANT VEL
QD(1,1)=0.0
QD(2,1)=QD(1,1)
QD(3,3)=G1.P1/(2.0A)
QD(3,5)=G2.P2/FACT2
QD(4,3)=QD(3,4)
QD(5,3)=QD(3,5)
QD(5,4)=QD(5,4)
QD(5,5)=QD(5,5)
100 C LOOP 2 15 FOR DUPLICATING THE SIMILAR PORTIONS OF AD
DO 2 I=3,5
DO 2 J=3,5
QD(I+3,J+3)=QD(I,J)
2 CONTINUE
C SET QDAR TO NONZERO PARTITION OF AD
<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1</td>
<td>I</td>
<td>65 68</td>
<td>149</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>J</td>
<td>66 68</td>
<td>38</td>
<td>INSTACK</td>
</tr>
<tr>
<td>177</td>
<td>2</td>
<td>I</td>
<td>110 113</td>
<td>129</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>203</td>
<td>2</td>
<td>J</td>
<td>111 113</td>
<td>38</td>
<td>INSTACK</td>
</tr>
<tr>
<td>212</td>
<td>43</td>
<td>I</td>
<td>115 117</td>
<td>158</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>220</td>
<td>43</td>
<td>J</td>
<td>116 117</td>
<td>38</td>
<td>INSTACK</td>
</tr>
<tr>
<td>231</td>
<td>44</td>
<td>I</td>
<td>120 122</td>
<td>128</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>235</td>
<td>44</td>
<td>J</td>
<td>121 122</td>
<td>29</td>
<td>INSTACK</td>
</tr>
<tr>
<td>244</td>
<td>45</td>
<td>I</td>
<td>123 125</td>
<td>158</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>252</td>
<td>45</td>
<td>J</td>
<td>124 125</td>
<td>38</td>
<td>INSTACK</td>
</tr>
<tr>
<td>262</td>
<td>3</td>
<td>I</td>
<td>127 129</td>
<td>128</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>266</td>
<td>3</td>
<td>J</td>
<td>128 129</td>
<td>28</td>
<td>INSTACK</td>
</tr>
</tbody>
</table>

**STATISTICS**

**PROGRAM LENGTH**

5558     365

52000B CM USED
SUBROUTINE PROP

SUBROUTINE PROP(PHIT,QDROOT,H,XT,YT,N,M)
REAL PHIT(1:N),QDROOT(1:N),XT(1:N),YT(1:N),H(1:N)
REAL TEMP1(1:N),TEMP2(1:N)

THIS ROUTINE IMPLEMENTS THE STATE TRANSITION EQUATION.

\[ XT(1) = PHIT \times XT(1) + QDROOT \times WD \]

WHERE

- \( XT \) = STATE VECTOR (NX1)
- \( PHIT \) = STATE TRANSITION MATRIX (NXN)
- \( QDROOT \) = STATE UNCERTAINTY COVARIANCE MATRIX (NXN)
- \( WD \) = GAUSSIAN DISTRIBUTED NOISE VECTOR (NX1)

AND THE OUTPUT EQUATION

\[ YT = H \times XT \]

WHERE

- \( YT \) = MEASURABLE OUTPUT VECTOR (MX1)
- \( H \) = STATE TO OUTPUT MAXTRIX (MXN)

CALL NOISE(TEMP1,1)
CALL MULT(QDROOT,TEMP1,1,1,TEMP2)
CALL MULT(PHIT,XT,1,1,1,TEMP1)
DO 1 I=1,N
  XT(I)=TEMP1(I,1)+TEMP2(I,1)
1...
ADD DETERMINISTIC VELOCITY OF 3 PIXELS OVER ANY TIME HISTORY IN THE X DIRECTION

XT(1,1)=XT(1,1)+15
CALL MULT(M,XT,1,1,1,1,YT)
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 PROP

VARIABLES   SN   TYPE   RELOCATION   RELocation
0   M   REAL   ARRAY   F.P.   100   I   INTEGER
0   N   INTEGER   ARRAY   F.P.   0   N   INTEGER   F.P.
0   PHIT   REAL   ARRAY   F.P.   0   QDROOT   REAL   ARRAY   F.P.
101   TEMP1   REAL   ARRAY   F.P.   111   TEMP2   REAL   ARRAY   F.P.
0   XT   REAL   ARRAY   F.P.   0   YT   REAL   ARRAY   F.P.

EXTERNALS   TYPE   ARGS
MULT   6   NOISE   2

STATEMENT LABELS
0   1
SUBROUTINE IDEAL(74/74 OPT=1 PNDMP F1T 4.B+528 09/21/81 12.15.51 PAGE 1)

SUBROUTINE IDEAL(REAL IMAX, S, XMAX, YMAX, N, X, Y, DATA, DX, DY)
REAL IMAX(3), XMAX(3), YMAX(3), S(12)
COMPLEX DATA(N, N), DX(N, N), DY(N, N)
DO 10 I = 1, N
DO 10 J = 1, N
DATA(I, J) = 0.
DX(I, J) = 0.
DY(I, J) = 0.
CONTINUE
10 IF(N.LT.8) N=8
IF((N-B).LT.2, LT, NZ) NZ=(N-B)/2
LM=2*N
LP=N-NZ
IB=(N-B)/2+1
DO 1 I = LM, LP
DO 1 J = L4, LP
X1 = X*FLOAT(J-IB)-XMAX(1)+.5
X2 = X*FLOAT(J-IB)-XMAX(2)+.5
X3 = X*FLOAT(J-IB)-XMAX(3)+.5
Y1 = Y*FLOAT(J-IB)-YMAX(1)+.5
Y2 = Y*FLOAT(J-IB)-YMAX(2)+.5
Y3 = Y*FLOAT(J-IB)-YMAX(3)+.5
ARG1 = (-5*((X1**2*S(11))+(X1+Y1)*S(2)+S(3))+(Y1**2*S(4)))
ARG2 = (-5*((X2**2*S(5))+(X2+Y2)*S(6)+S(7))+(Y2**2*S(8)))
ARG3 = (-5*((X3**2*S(3))+(X3+Y3)*S(10)+S(11))+(Y3**2*S(12)))
DATA(I, J) = IMAX(1)*EXP(ARG1)+IMAX(2)*EXP(ARG2)+IMAX(3)*EXP(ARG3)
ARG4 = X1*S(11)-S(2)+S(3))
ARG5 = X2*S(5)-S(6)+S(7))
ARG6 = X3*S(3)-S(10)+S(11))
ARG7 = Y1*S(4)-S(2)+S(3))
ARG8 = Y2*S(6)-S(7))
ARG9 = Y3*S(12)-S(10)+S(11))
DX(I, J) = ARG4*IMAX(1)*EXP(ARG1)+ARG5*IMAX(2)*EXP(ARG2)
DY(I, J) = ARG7*IMAX(1)*EXP(ARG1)+ARG8*IMAX(2)*EXP(ARG2)
1 CONTINUE
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 IDEAL

VARIABLES SN TYPE RELOCATION
24-6 ARG1 REAL 257 ARG2 REAL
24-0 ARG3 REAL 261 ARG4 REAL
24-6 ARG5 REAL 263 ARG6 REAL
24-1 ARG7 REAL 265 ARG8 REAL
24-6 ARG9 REAL 0 DATA COMPLEX ARRAY F.P.
0 DX COMPLEX ARRAY F.P.
2-3 I INTEGER 247 IB INTEGER
<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
<th>RELLOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 IMAK</td>
<td>REAL ARRAY F.P.</td>
<td>244 J</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td>245 LMA</td>
<td>INTEGER</td>
<td>246 LP</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td>0 N</td>
<td>INTEGER</td>
<td>0 N2</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td>0 S</td>
<td>REAL ARRAY F.P.</td>
<td>0 X</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>0 XMAX</td>
<td>REAL ARRAY F.P.</td>
<td>250 X1</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>251 X2</td>
<td>REAL</td>
<td>252 X3</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>0 Y</td>
<td>REAL</td>
<td>0 YMAX</td>
<td>REAL ARRAY F.P.</td>
<td></td>
</tr>
<tr>
<td>253 Y1</td>
<td>REAL</td>
<td>254 Y2</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>255 Y3</td>
<td>REAL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXTERNALS</th>
<th>TYPE</th>
<th>ARGS</th>
<th>LIBRARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP REAL</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INLINE FUNCTIONS</th>
<th>TYPE</th>
<th>ARGS</th>
<th>INTRIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOAT REAL</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATEMENT LABELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 10</td>
<td>I</td>
<td>4 9</td>
<td>16B</td>
<td>NOT INNER</td>
<td></td>
</tr>
<tr>
<td>20 10</td>
<td>J</td>
<td>5 9</td>
<td>4B</td>
<td>INSTACK</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>45 1</td>
<td>I</td>
<td>15 37</td>
<td>172B</td>
<td>EXT REFS</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>47 1</td>
<td>J</td>
<td>16 37</td>
<td>165B</td>
<td>EXT REFS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM LENGTH</td>
</tr>
<tr>
<td>52000B CM USED</td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
</table>
SUBROUTINE SMOOTH

SUBROUTINE SMOOTH(DATA,SDATA,ALPHA,N,ITERA)
COMPLEX DATA(N,N),SDATA(N,N)
C THIS ROUTINE SMOOTH'S RAW DATA ARRAY DATA USING EXPONENTIAL
C SMOOTHING. WEIGHTING FACTOR ALPHA IS USED TO GENERATE THE
C SMOOTHED DATA IN ARRAY SDATA. THE PARAMETER ITERATION IS
C USED TO DETERMINE THE WEIGHTING FACTOR WHEN FEWER THEN
C 1/ALPHA ITERATIONS HAVE BEEN DONE.
C A=1./ITERA
IF(A.LT.ALPHA) A=ALPHA
10 1 DO 3 I=1,N
DO 3 J=1,N
3 SDATA(I,J)=A*DATA(I,J)+(1.-A)*SDATA(I,J)
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 SMOOTH

VARIABLES

<table>
<thead>
<tr>
<th>N</th>
<th>TYPE</th>
<th>RELOCATION</th>
<th>0</th>
<th>ALPHA</th>
<th>REAL</th>
<th>F.P.</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>A</td>
<td>REAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>DATA</td>
<td>COMPLEX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ITERA</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STATEMENT LABELS

| 1 | INACTIVE | 0 | 3 |

LOOPS

<table>
<thead>
<tr>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
<td>12</td>
<td>23B</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>12</td>
<td>68</td>
</tr>
</tbody>
</table>

STATISTICS

<table>
<thead>
<tr>
<th>PROGRAM LENGTH</th>
<th>56B</th>
<th>46</th>
</tr>
</thead>
</table>
SUBROUTINE NOISE  74/74  OPT=1  PMDCMP  

1  SUBROUTINE NOISE(W,N)  
   REAL W(N)  
   C  THESE STATEMENTS WERE USED IN THE MODCOMP VERSION OF THE SOFTWARE  
   C  DATA IFIRST/0/  
   5  C  IA=IA2345  
   C  IFIRST=1  
   IA=1  
   10  DO 2 I=1,N  
   1 CALL GAUSS(IA,1Y,VAL)  
   W(I)=VAL  
   IA=IY  
   2  CONTINUE  
   RETURN  
   END  

SYMBOLIC REFERENCE MAP (R=1)  

ENTRY POINTS  
3  NOISE  

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
<th>24</th>
<th>IA</th>
<th>INTEGER</th>
<th>0</th>
<th>N</th>
<th>INTEGER</th>
<th>F.P.</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 I</td>
<td></td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 1Y</td>
<td></td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27 VAL</td>
<td></td>
<td>REAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXTERNALS TYPE ARGS  
GAUSS 3  

STATEMENT LABELS  
0 1  INACTIVE  0 2  

<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
<th>EXT REFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 2</td>
<td>1</td>
<td>9 13</td>
<td>70</td>
<td></td>
<td>EXT</td>
<td></td>
</tr>
</tbody>
</table>

STATISTICS  
| PROGRAM LENGTH | 338 | 27 |  
| 520000 CM USED |   |   |  

SUBROUTINE INPUT3  74/74  OPT=1  PMOMP

SUBROUTINE INPUT3(IMAX,S,XMAX,YMAX,N,X,Y,DATA,CENX,CENY)
REAL IMAX(3),S(12),XMAX(3),YMAX(3)
COMPLEX DATA(N,N)
THIS ROUTINE DETERMINES REAL MODEL INTENSITY AND CENTROID
VALUES FOR AN 8X8 PIXEL FOV. ZERO PADDING IS ACCOMPLISHED BY
CENTERING THE 8X8 PIXEL FOV WITHIN A NULL NXN SPACE.
THE COORDINATE SYSTEM IS AS DEFINED BELOW:

(0.0)........................

\*
GAUSSIAN
\*

(X,Y).........

FOV
n : n

THE INTENSITY PATTERN IS DEFINED TO BE 3 GAUSSIAN DISTRIBUTIONS
OF INTENSITY IMAX(I),I=1,3 LOCATED AT XMAX(I),YMAX(I),I=1,3
WITH COVARIANCE S(I),I=1,3. THE UPPER LEFT CORNER OF THE 8X8 PIXEL
FOV IS DEFINED TO BE LOCATION X,Y MICROGRAD.
THE INTENSITY AT EACH PIXEL IS DETERMINED BY INTEGRATING THE
INTENSITY OF 25 EQUALLY SPACED SPOTS WITHIN THE PIXEL.

ZERO OUT FOV SPACE.
DO 10 I=1,N
DO 10 J=1,N
DATA(I,J)=0.
SUMX=0.
SUMY=0.
SUMAVG=0.
LM=N/2-3
LP=N/2+4
D3 1 I=LM,LP
DO 2 J=LM,LP
AVG=0.
DIVIDE PIXEL I,J INTO 25 SEGMENTS
D0 3 K1=1.5
D0 4 K2=1.5
DELY=(I-3)+1.0+(K1-1)*.2
DELX=(J-3)+1.0+(K2-1)*.2
XP=X*CELX
YP=Y*CELX
X1=XP-XMAX(1)
Y1=YP-YMAX(1)
X2=XP-XMAX(2)
Y2=YP-YMAX(2)
X3=XP-XMAX(3)
Y3=YP-YMAX(3)
ARG1=-5*((X1++2*S(1))+(X1+Y1)*(S(2)+S(3))+(Y1+2*S(4))))
C

WRITE(6,101) X1,Y1,S(1),S(2),S(3),S(4),IMAX(1),ARG1

101 FORMAT(2X,6E12.5)

ARG2=-.5*((X2+2*S(5))+X2)*X2+Y2))
ARG3=-.5*(((X3+2*S(6))+X3)*X3+Y3))
FXY=IMAX(1)*EXP(ARG1)+IMAX(2)*EXP(ARG2)+IMAX(3)*EXP(ARG3)

AVG=AVG+FXY
SUMX=SUMX+FXY
SUMY=SUMY+FXY
CONTINUE
DATA(1,J)=AVG/25.
C
WRITE(6,100) I,J,DATA(I,J)

100 FORMAT(2X,2I4,2X,E12.5)
SUMAVG=SUMAVG+AVG
CONTINUE
1
CENX=SUMX/SUMAVG
CENY=SUMY/SUMAVG
RETURN

END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 INPUT3

VARIABLES  SN  TYPE  RELOCATION
221  ARG1  REAL  222  ARG2  REAL
223  ARG3  REAL  204  AVG  REAL
0  CENX  REAL  0  CENY  REAL
0  DATA  COMPLEX  ARRAY  0  S  REAL
207  DELY  REAL  210  DELX  REAL
175  I  INTEGER  0  IMAX  REAL
175  J  INTEGER  205  K1  INTEGER
206  X2  INTEGER  202  LM  INTEGER
203  LP  INTEGER  0  N  INTEGER
0  S  REAL  201  SUMAVG  REAL
177  SUMX  REAL  200  SUMY  REAL
0  X  REAL  0  XMAX  REAL
211  XP  REAL  213  X1  REAL
215  X2  REAL  217  X3  REAL
0  Y  REAL  0  YMAX  REAL
212  YP  REAL  214  Y1  REAL
216  Y2  REAL  220  Y3  REAL

EXTERNALS  TYPE  ARGS
EXP  REAL  'LIBRARY

STATEMENT LABELS
0  1  0  2  0  3
0  3  0  4
16  100  FMT  NO REFS

114  101  FMT  NO REFS
SUBROUTINE DISPLAY  (IXSIZE, IYSIZE, DATA)
   DIMENSION IXY(122)
   COMPLEX DATA(IXSIZE, IYSIZE)
   WRITE(6,102)
   FORMAT(1H1)
      NUM=IXSIZE+2
   DO 5 I=1, NUM
      IXY(I)=1H-
      WRITE(6,103) (IXY(I), I=1, NUM)
   10 FORMAT(T4,122A1)
      DMIN=1.530
      DMAX=-1.530
      DO 1 I=1, IYSIZE
         DMAX=MAX1(DMAX, CABS(DATA(J, I)))
      15 DMIN=MIN1(DMIN, CABS(DATA(J, I)))
      DO 4 J=1, IXSIZE
         IXY(J)=1H-
      20 X=(CABS(DATA(J, I))-DMIN)/(DMAX-DMIN)
         IF(X.GT.-.92) IXY(J)=1H0
         IF((X.GT.-.28).AND.(X.LE.-.42)) IXY(J)=1HX
         IF((X.GT.-.14).AND.(X.LE.-.28)) IXY(J)=1H-
      25 CONTINUE
      NUM=IXSIZE+1
      IXY(NUM)=1H-
      WRITE(6,100) (IXY(I), I=1, NUM)
   100 FORMAT(T4,122A1)
      DO 3 I=1, IXSIZE
         IXY(I)=1H-
         X=(CABS(DATA(J, I))-DMIN)/(DMAX-DMIN)
         IF((X.GT.-.56).AND.(X.LE.-.70)) IXY(I)=1H-
         IF((X.GT.-.70).AND.(X.LE.-.84)) IXY(I)=1H+
         IF((X.GT.-.84)) IXY(I)=1H#
      35 CONTINUE
      WRITE(6,101) (IXY(I), I=1, IXSIZE)
   101 FORMAT(1H+, T5, 120A1)
      CONTINUE
      NUM=IXSIZE+2
   40 DO 6 I=1, NUM
      IXY(I)=1H-
      WRITE(6,103) (IXY(I), I=1, NUM)
   45 RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 DISPLAY
<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
<th>DATA</th>
<th>COMPLEX</th>
<th>ARRAY</th>
<th>F.P.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMIN</td>
<td>0</td>
<td>REAL</td>
<td>262 DMAX</td>
<td>260</td>
<td>I</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td>IXSIZE</td>
<td>0</td>
<td>INTEGER</td>
<td>263 J</td>
<td>265</td>
<td>IXY</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td>NUM</td>
<td>257</td>
<td>INTEGER</td>
<td>264 X</td>
<td></td>
<td></td>
<td>REAL</td>
<td></td>
</tr>
</tbody>
</table>

FILE NAMES
- TAPE6
- MODE
- FMT

EXTERNALS
- CADS
- REAL
- 1 LIBRARY

INLINE FUNCTIONS
- AMAX1
- REAL
- 0 INTRIN
- AMIN1
- REAL
- 0 INTRIN

STATEMENT LABELS
- 0 1
- 0 2
- 0 4
- 0 5
- 221 100 FMT
- 231 101 FMT
- 206 102 FMT
- 206 102 FMT
- NO REFS

LOOPS
<table>
<thead>
<tr>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>5</td>
<td>7-8</td>
<td>33</td>
<td>INSTACK</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>13-16</td>
<td>245</td>
<td>EXT REF</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>14-16</td>
<td>218</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>57</td>
<td>4</td>
<td>17-38</td>
<td>1108</td>
<td>EXT REF</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>18-24</td>
<td>345</td>
<td>NOT INNER</td>
</tr>
<tr>
<td>124</td>
<td>3</td>
<td>29-35</td>
<td>329</td>
<td>EXT REF</td>
</tr>
<tr>
<td>173</td>
<td>6</td>
<td>40-41</td>
<td>38</td>
<td>INSTACK</td>
</tr>
</tbody>
</table>

STATISTICS
- PROGRAM LENGTH: 4738
- 315 CM USED
- 520098 CM USED
SUBROUTINE SHIFT

COMPLEX DATA(N,N), TEMP1, TEMP2, TEMP3, TEMP4, FX, FXC, FY, FYC

IN THE FREQUENCY DOMAIN, THE ARRAY DATA IS ASSUMED TO BE AN NXN ARRAY OF FOURIER TRANSFORM COMPONENTS AS GENERATED BY THE TRANSFORM ROUTINE FOURT. FOR EXAMPLE, FOR A 6X6 ARRAY DATA

PHASE SHIFTING IS IMPLEMENTED BY MULTIPLYING THE FOURIER TRANSFORM COMPONENTS BY

\[ \exp(j2\pi(fx \cdot xshift + fy \cdot yshift)) \]

XSHIFT AND YSHIFT ARE THE SHIFTS IN THE X AND Y COORDINATE DIRECTIONS.

PI = 3.141592654

DEM = FLOAT(N)
NCENT = N/2 + 1
DO 1 I = 1, NCENT
1 CONTINUE

FX = CMPLX(0., -2.*PI*(J-1)*XSHIFT/DEM)
FY = CMPLX(0., -2.*PI*(I-1)*YSHIFT/DEM)
FXC = CONJG(FX)
FYC = CONJG(FY)
TEMP1 = DATA(I, J)

DATA(I, J) = TEMP1*CEXP(FX + FY)
IF(J.EQ.1 .OR. J.EQ.NCENT) GO TO 10
IF(J.EQ.1 .OR. J.EQ.NCEN + 2) GO TO 20
TEMP2 = DATA(I, N+2-J)
TEMP3 = DATA(N+2-I, J)
TEMP4 = DATA(N+2-I, N+2-J)
DATA(I, N+2-J) = TEMP2*CEXP(FXC + FY)
DATA(N+2-I, J) = TEMP3*CEXP(FX + FYC)
DATA(N+2-I, N+2-J) = TEMP4*CEXP(FXC + FYC) GO TO 1

IF(J.EQ.1 .OR. J.EQ.NCENT) GO TO 10
TEMP2 = DATA(I, N+2-J)
DATA(I, N+2-J) = TEMP2*CEXP(FXC + FY)
GO TO 1

DATA(N+2-I, J) = TEMP3*CEXP(FX + FYC)
SUBROUTINE SHIFT 74/74 OPT=1 PMDMP
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 SHIFT

VARIABLES SN TYPE RELLOCATION
0 DATA COMPLEX ARRAY F.P. 242 DEM REAL
231 FX COMPLEX 233 FXC COMPLEX
235 FY COMPLEX 237 FYC COMPLEX
234 I INTEGER 245 J INTEGER
0 N INTEGER F.P. 243 NCNT INTEGER
241 PI REAL 221 TEMPI COMPLEX
223 TEM1 COMPLEX 225 TEMPI COMPLEX
227 TEMP REAL 0 XSHIFT REAL F.P.

EXTERNALS TYPE ARG'S LIBRARY
CEXP COMPLEX 1

INLINE FUNCTIONS TYPE ARG'S
CMPLX COMPLEX 2 INTRIN
FLOAT REAL 1 INTRIN

STATEMENT LABELS 0 7 INACTIVE 137 10
210 1 166 20

LOOPS LABEL INDEX FROM-TO LENGTH PROPERTIES
15 1 I 33 56 1778 EXT REFS NOT INNER
17 1 J 34 56 1748 EXT REFS

STATISTICS
PROGRAM LENGTH 2758 189
52000 BY DM USED
SUBROUTINE DERIV

SUBROUTINE DERIV(DATA,N,DX,DY)
COMPLEX DATA(N,N),TEMP1,TEMP2,TEMP3,TEMP4,FX,FXC,FY,FYC
COMPLEX DX(N,N),DY(N,N)

THIS ROUTINE IMPLEMENTS SPATIAL PARTIAL DERIVATIVES
IN THE FREQUENCY DOMAIN. THE ARRAY DATA IS ASSUMED TO BE THE
N X N ARRAY OF FOURIER TRANSFORM COMPONENTS AS GENERATED BY THE
TRANSFORM ROUTINE FOURI. FOR EXAMPLE, FOR A 6 X 6 ARRAY DATA

<table>
<thead>
<tr>
<th>X0 Y0</th>
<th>X1 Y0</th>
<th>X2 Y0</th>
<th>X3 Y0</th>
<th>X* Y0</th>
<th>X* Y0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0 Y1</td>
<td>X1 Y1</td>
<td>X2 Y1</td>
<td>X3 Y1</td>
<td>X2 Y1</td>
<td>X1 Y1</td>
</tr>
<tr>
<td>X0 Y2</td>
<td>X1 Y2</td>
<td>X2 Y2</td>
<td>X3 Y2</td>
<td>X2 Y2</td>
<td>X1 Y2</td>
</tr>
<tr>
<td>X0 Y3</td>
<td>X1 Y3</td>
<td>X2 Y3</td>
<td>X3 Y3</td>
<td>X2 Y3</td>
<td>X1 Y3</td>
</tr>
<tr>
<td>X0 Y4</td>
<td>X1 Y4</td>
<td>X2 Y4</td>
<td>X3 Y4</td>
<td>X2 Y4</td>
<td>X1 Y4</td>
</tr>
</tbody>
</table>

DIFFERENTIATION IS IMPLEMENTED BY MULTIPLYING THE
FOURIER TRANSFORM COMPONENTS BY
J+2*PI*FX AND J+2*PI*FY

PI=3.141592654
DEM=FLOAT(N)
NCENT=N/2+1
DO 1 J=1,NCENT
DO 1 D=1,NCENT
FX=CMPLX(0.,+2.*PI*(J-1)/DEM)
FX=CMPLX(0.,+2.*PI*(I-1)/DEM)
FXC=CONJG(FX)
TEMP1=DATA(I,J)
DX(I,J)=TEMP1+FX
DY(I,J)=TEMP1+FY
IF(I.EQ.1) GO TO 10
IF(J.EQ.1.OR.J.EQ.NCEN) GO TO 20
TEMP2=DATA(I,N+2-J)
TEMP3=DATA(N+2-I,J)
TEMP4=DATA(N+2-I,N+2-J)
DX(I,N+2-J)=TEMP2+FX
DY(I,N+2-J)=TEMP2+FY
DX(N+2-I,J)=TEMP3+FX
DY(N+2-I,N+2-J)=TEMP4+FXC
GO TO 1
10 IF(J.EQ.1.OR.J.EQ.NCEN) GO TO 1
TEMP2=DATA(I,N+2-J)
DX(I,N+2-J)=TEMP2+FXC
DY(I,N+2-J)=TEMP2+FY
GO TO 1
20 TEMP3=DATA(N+2-I,J)
SUBROUTINE DERIV  74/74  OPT=1  PM010P

DX(N+2-1,J)=TEMP3*FX
DY(N+2-1,J)=TEMP3*FYC
60 1 CONTINUE
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3  DERIV

VARIABLES  SN  TYPE   RELLOCATION
           0  DATA  COMPLEX  ARRAY  F.P.     225  DEM  REAL
           0  DX   COMPLEX  ARRAY  F.P.     0  DY   COMPLEX  ARRAY  F.P.
    214  FX   COMPLEX
    220  FY   COMPLEX
    227  J    INTEGER
    0  N    INTEGER  F.P.     226  NCENT  INTEGER
    224  PI   REAL
    206  TEMP  COMPLEX
    212  TEMP  COMPLEX

INLINE FUNCTIONS  TYPE   ARGS
                COMPLEX  2  INTRIN  CONJG  COMPLEX  1  INTRIN
                FLOAT   REAL  1  INTRIN

STATEMENT LABELS
162  1  0  7  INACTIVE  124  10
146  20

LOOPS  LABEL  INDEX  FROM-TO  LENGTH  PROPERTIES
16  1  I   31  60  1510  NOT INNER
36  1  J   32  60  1260  OPT

STATISTICS
PROGRAM LENGTH  260B  176
52000B CM USED
SUBROUTINE FOUR(DATA,NN,NDIM,ISIGN,IFORM,WORK) FOR INFORMATION CONTACT MR. MARK HALLER 4950/ADDS/56248

THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN

TRANSFORM(K1,K2,...) = SUM(DATA(J1,J2,...)*EXP(ISIGN*2*PI*SORT(-1)*(J1-1)*(K1-1)/NN(1)+J2-1)*NN(2)+...)), SUMMED FOR ALL J1, K1 BETWEEN 1 AND NN(1), J2, K2 BETWEEN 1 AND NN(2), ETC.

THERE IS NO LIMIT TO THE NUMBER OF SUBSCRIPTS. DATA IS A MULTIDIMENSIONAL COMPLEX ARRAY WHOSE REAL AND IMAGINARY PARTS ARE ADJACENT IN STORAGE, SUCH AS FORTRAN IV PLACES THEM.

IF ALL IMAGINARY PARTS ARE ZERO (DATA ARE DISGUISED REAL), SET IFORM TO ZERO TO CUT THE RUNNING TIME BY UP TO FORTY PERCENT.

IF OTHERWISE, IFORM = +1. THE LENGTHS OF ALL DIMENSIONS ARE STORED IN ARRAY NN, OF LENGTH NDIM. THEY MAY BE ANY POSITIVE INTEGERS, AND THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS, ESPECIALLY FAST ON NUMBERS RICH IN FACTORS OF TWO. ISIGN IS +1 OR -1. IF A -1 TRANSFORM IS FOLLOWED BY A +1 ONE (OR A +1 BY A -1) THE ORIGINAL DATA REAPPEAR, MULTIPLIED BY NTO(T(+NN(1)*NN(2)*...)). TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED IN ARRAY WORK.

DIMENSIONS ARE NOT POWERS OF TWO, ARRAY WORK MUST BE SUPPLIED.

COMPLEX OF LENGTH EQUAL TO THE LARGEST NON 2**K DIMENSION.

IF OTHERWISE, REPLACE WORK BY ZERO IN THE CALLING SUBRUTINE.

NORMAL FORTRAN DATA ORDERING IS EXPECTED, FIRST SUBRUTINE VARYING FASTER. ALL SUBSCRIPTS BEGIN AT ONE.

RUNNING TIME IS MUCH SHORTER THAN THE NAIVE NTO(T+2. BEING GIVEN BY THE FOLLOWING FORMULA. DECOMPOSE NTO(T INTO 2**K1+3**K2+... AND LET SUM=2*K1+3*K2+... THE TIME TAKEN BY A MULTI-DIMENSIONAL TRANSFORM ON THESE NTO(T DATA IS T = TO + NTO(T-1)*NTO(T)/1000000, ON THE CDC 3300 (FLOATING POINT ADD TIME IS 3.2 MICROSECONDS), T = 3000+NT/1000000+NT+6000+NT*(500+K1+SUM=500+K1+NT/1000000, ON CDC 750-NF)(MICROSECONDS). IN ADDITION, THE ACCURACY IS GREATLY IMPROVED, AND THE RMS RELATIVE ERROR IS BOUNDED BY 3.2**E(-B)*SUM(FACTORS(J)+1.5), WHERE B IS THE NUMBER OF BITS IN THE FLOATING POINT FRACTION AND FACTORS(J) ARE THE PRIME FACTORS OF NT.

PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES RACER. RALPH ALTER SUGGESTED THE IDEA FOR THE DIGIT REVERSAL. THIS IS THE FASTEST AND MOST VERSATILE VERSION OF THE FFT KNOWN TO THE AUTHOR. SHORTER PROGRAMS FOUR AND FOURZ RESTRICT DIMENSION LENGTHS TO POWERS OF TWO. SEE IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.
C 3. CONCEPTUALLY AT LEAST, THE INPUT DATA AND THE TRANSFORM OUTPUT
C REPRESENT SINGLE CYCLES OF PERIODIC FUNCTIONS.
C
C EXAMPLE 1. THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A
C COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV.
C DIMENSION DATA(32,25,13), WORK(50), NN(3)
C COMPLEX DATA
C DATA NN/32,25,13/
C DO 1 I=1,32
C DO 1 J=1,25
C DO 1 K=1,13
C 1 DATA(I,J,K) = COMPLEX VALUE
C CALL FOURT (DATA, NN, 3, -1, 1, WORK)
C
C EXAMPLE 2. ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF
C LENGTH 64 IN FORTRAN II.
C LENGTH 64 IN FORTRAN II.
C DIMENSION DATA(2,64)
C DO 2 I=1,64
C DATA(1,1) = REAL PART
C DATA(2,1) = 0
C CALL FOURT(DATA, 64, 1, -1, 0, 0)
C
C DIMENSION DATA(1), NN(1), IFACT(32), WORK(1)
C
C 6800 INITIALIZATION
C WR=0.
C WI=0.
C WSTP=0.
C WSTP1=0.
C T=0.
C IF(NOT=I1920,1,1)
C 1 NTOT=2
C DO 2 IDIM=1, NDIM
C IF(N(NIDIM))920,920,2
C 2 NTOT=NTOT*NN(IDIM)
C
C MAIN LOOP FOR EACH DIMENSION
C N3=2
C DO 910 IDIM=1, NDIM
C N=NN(IDIM)
C NP2=NP1+N
C IF(N) 1920,900,5
C 910 NTWO=NP1
C IF(F=1)
C DIV=2
C 10 IQ=DIV-M/IDIV
C I=1M=M-IDIV+IQ
C IF(I) 1950,11,11
C 11 IF(I5) 20,12,20
C 12 NTWO=NTWO+NTWO
C M=IQ
C GO TO 10
C 20 IDIV=3
<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>FORT 74/74</th>
<th>OPT=1</th>
<th>PPDMP</th>
<th>FTN 4.8+528</th>
<th>09/21/81</th>
<th>12.15.51</th>
<th>PAGE 3</th>
</tr>
</thead>
</table>

| 115 | 30 | IQOUT=M/IDIV |
|     |     | IREM=M-IDIV*IQOUT |
|     |     | IF(IQOUT-IDIV)60,31,31 |
|     |     | IF(IREM)40,32,40 |
|     |     | IFACT(IF)=IDIV |
|     | 120 | IF=IF+1 |
|     |     | M=IQOUT |
|     |     | GO TO 30 |
|     |     | IDIV=IDIV+2 |
|     |     | GO TO 30 |
|     | 125 | IF(IREM)60,51,60 |
|     |     | NTWO=NTWO+NTWO |
|     |     | GO TO 70 |
|     |     | IFACT(IF)=M |

130 C SEPARATE FOUR CASES:--
  1. COMPLEX TRANSFORM OR REAL TRANSFORM FOR THE 4TH, 5TH, ETC.
  DIMENSIONS.
  2. REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION. METHOD--
     TRANSFORM HALF THE DATA, SUPPLYING THE OTHER HALF BY CON-
     JUGATE SYMMETRY.
  3. REAL TRANSFORM FOR THE 1ST DIMENSION, N ODD. METHOD--
     TRANSFORM HALF THE DATA AT EACH STAGE, SUPPLYING THE OTHER
     HALF BY CONJUGATE SYMMETRY.
  4. REAL TRANSFORM FOR THE 1ST DIMENSION, N EVEN. METHOD--
     TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 VICE REAL PARTS
     ARE THE EVEN NUMBERED REAL VALUES AND WHOSE IMAGINARY PARTS
     ARE THE ODD NUMBERED REAL VALUES. SEPARATE AND SUPPLY
     THE SECOND HALF BY CONJUGATE SYMMETRY.

140 | 70 | NON2=NP1*(NP2/NTWO) |
|     |     | ICASE=1 |
|     |     | IF(IDIM-4)71,90,90 |
|     | 71 | IF(IDIM4)72,72,90 |
|     |     | ICASE=2 |
|     |     | IF(IDIM-1)73,73,90 |
|     | 73 | ICASE=3 |
|     |     | IF(NTWO-NP1)90,90,74 |
|     |     | ICASE=4 |
|     |     | NTWO=NTWO/2 |
|     |     | N=N/2 |
|     |     | NP2=NP2/2 |
|     |     | NTOT=NTOT/2 |
|     |     | i=3 |
|     |     | DD 80 J=2,NTOT |
|     | 160 | DATA(J)=DATA(I) |
|     |     | i=i+2 |
|     | 90 | IANG=NP1 |
|     |     | IF(IFCASE-2)100,95,100 |
|     |     | IANG=NP0*(1+NPREV2) |

165 C SHUFFLE ON THE FACTORS OF TWO IN N. AS THE SHUFFLING
     CAN BE DONE BY SINGLE INTERCHANGE, NO WORKING ARRAY IS NEEDED

170 | 100 | IF(NTWO-NP1)600,600,110 |
|     | 110 | NP2HF=NP2/2 |
|     |     | j=j+1 |
SUBROUTINE FOURT

74/74 OPE=1 PMOMP

FTN 4.8+52B

09/21/81 12.15.51 PAGE 4

DO 150 12=1,NP2,NON2
IF(J-12)120,130,130
120 I=MAX+12+NON2-2
DO 125 I=12,11MAX+2
DO 125 I=12,NOD2,11MAX+2
J=J+13-12
TEMP=DATA(I3)
DATA(I3)=DATA(I3+1)
DATA(I3+1)=DATA(I3)
DATA(I3)=TEMP
125 DATA(J3+1)=TEMP
130 M=NP2HF
140 IF(J-M)150,150,145
145 J=J+M
M=M/2
IF(M-NON2)150,140,140
150 J=J+M
190 C

MAIe LOOP FOR FACTORS OF TWO. PERFORM FOURIER TRANSFORMS OF
LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED. THE TWIDDLE FACTO:

W=EXP(ISIGN+2*PI*SORT(-1)+M/(4*MMAX)). CHECK W=SIGN+QRT(-1)*CONJUGATE(W).
195 C

NON2=NON2+NON2
IFAR=INTQ/NP1
310 IFAR=IFAR/4
320 IF=IFAR/4
330 DO 300 I=11,11NRG,9
DO 300 J=11,NON2,NP1
DO 300 M=11,11NOD2,11NOD2
K2=K+NON2+1
300 DATA(K2)=DATA(K2)
DATA(K)=DATA(K+1)+TEMP
DATA(K+1)=DATA(K)+TEMP
310 DATA(K+1)=DATA(K+1)+TEMP
320 DATA(K+1)=DATA(K+1)+TEMP
350 MM2=MMAX+NON2
360 IF(1+MMAX+NP2HF)1370,600,600
370 LU-MAX+1NON2,NMAX+2
380 IF(MAX+NON2)105,405,390
380 THETA=THETA+FLOAT(1+NON2)/FLOAT(4+MMAX)
390 IF(ISIGN+100,290,390
390 SIGMA=SIGMA
400 WR=COS(THETA)
410 SI=SIGNI THETA
420 WR2=WR+W21+W1
430 WR2=W22*W21+W1
440 DO 570 1=NON2,1G,410
570 IF(MAX+NON2)100,420,410
580 WR2=WR+W1
590 W22=W22+W21+W1
600 W31=W31+W21+WR
6101-1
<table>
<thead>
<tr>
<th>SUBROUTINE FOUR</th>
<th>74/74</th>
<th>OPT=1</th>
<th>PMQMP</th>
<th>FTN 4.8+52B</th>
<th>09/21/81 12.15.51</th>
<th>PAGE 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>640</td>
<td>THETA=TWOP1/FLOAT(IFACT(IF))</td>
<td>FFTT3320</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>645</td>
<td>THETA=THETA</td>
<td>FFTT3330</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>650</td>
<td>SINH=1N/THETA/2.</td>
<td>FFTT3340</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WSTPH=2.*SINH+SINTH</td>
<td>FFTT3350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WSTPI=1N(THETA)</td>
<td>FFTT3360</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>350</td>
<td>KSTEPZ2.*IFACT(IF)</td>
<td>FFTT3370</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>KRAK=KST=1NFACT(IF)/2+1</td>
<td>FFTT3380</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 690 IT=1,119N,2</td>
<td>FFTT3390</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 690 IT=1,119N,2</td>
<td>FFTT3400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 690 KMIN=1,KRAK,KSTEP</td>
<td>FFTT3410</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>365</td>
<td>JMAX=J1+1N9N=1F1P1</td>
<td>FFTT3420</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2MAX=J1+1F2+1P1</td>
<td>FFTT3430</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J3MAX=J1+1F2+1P1</td>
<td>FFTT3440</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 690 J3=J1,3MAX,NP1</td>
<td>FFTT3450</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2MAX=J1+1F2+1P2</td>
<td>FFTT3460</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUM1=NMIN+J1*(J1+13)/IFACT(IF)/NP1+HF</td>
<td>FFTT3470</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>365</td>
<td>IF(NMIN+1)655,655,655</td>
<td>FFTT3480</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>SUMR=0.</td>
<td>FFTT3490</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUM1=0.</td>
<td>FFTT3500</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 660 J2=J3,2MAX,1F2</td>
<td>FFTT3510</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUMR+SUMR+DATA(J2)</td>
<td>FFTT3520</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>365</td>
<td>SUM1+SUMR+DATA(J2+1)</td>
<td>FFTT3530</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K)=SUMR</td>
<td>FFTT3540</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K+1)=SUMI</td>
<td>FFTT3550</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 680</td>
<td>FFTT3560</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>370</td>
<td>KCONJ=K+2*(N-KMIN+1)</td>
<td>FFTT3570</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2=J2MAX</td>
<td>FFTT3580</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUMR+DATA(J2)</td>
<td>FFTT3590</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUM1+DATA(J2+1)</td>
<td>FFTT3600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD3R=0.</td>
<td>FFTT3610</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD3I=0.</td>
<td>FFTT3620</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>375</td>
<td>J2=J2+1F2</td>
<td>FFTT3630</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMP=R+SUMR</td>
<td>FFTT3640</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPI=SUMI</td>
<td>FFTT3650</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUMR=SUMR+SUMR+OLD3R+DATA(J2)</td>
<td>FFTT3660</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUM1=SUMR+SUMI+OLD3I+DATA(J2+1)</td>
<td>FFTT3670</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>380</td>
<td>OLD3R+TEMPR</td>
<td>FFTT3680</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD3I+TEMPI</td>
<td>FFTT3690</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2=J2+1F2</td>
<td>FFTT3700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF(J2=J3)675,675,670</td>
<td>FFTT3710</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>385</td>
<td>TEMP=R+SUMR+OLD3R+DATA(J2)</td>
<td>FFTT3720</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMPI=1*I+SUMI</td>
<td>FFTT3730</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K)=TEMPR+TEMPI</td>
<td>FFTT3740</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K)=TEMPR+TEMPI</td>
<td>FFTT3750</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMP=WR+SUMR+OLD3I+DATA(J2+1)</td>
<td>FFTT3760</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMP=WR+SUMR</td>
<td>FFTT3770</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEMP=1*I+SUMR</td>
<td>FFTT3780</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K+1)=TEMPR+TEMPI</td>
<td>FFTT3790</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WORK(K+1)=TEMPR+TEMPI</td>
<td>FFTT3800</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>390</td>
<td>CONTINUE</td>
<td>FFTT3810</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF(K=N-1)685,685,686</td>
<td>FFTT3820</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WR=Wr+STPR+1.</td>
<td>FFTT3830</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>395</td>
<td>WR=Wr+STPR+1.</td>
<td>FFTT3840</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 600</td>
<td>FFTT3850</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>TEMP=WR</td>
<td>FFTT3860</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WR=Wr+STPR+WI+STPI+WR</td>
<td>FFTT3870</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WI=TEMPR+STPI+WI 1STPR+WI</td>
<td>FFTT3880</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBROUTINE FOUT 74/74</td>
<td>OPT=1 PMDMP</td>
<td>FTN 4.8+52B</td>
<td>09/21/81 12.15.51</td>
<td>PAGE 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-----------------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>400 690</td>
<td>TWDWR=WR+WR</td>
<td>FTTT3890</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>691 IF(1P1-NP2)695,692</td>
<td>FTTT3900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>692 K=1</td>
<td>FTTT3910</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12MAX=13+NP2=NP1</td>
<td>FTTT3920</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>405 DD 603 12=13,12MAX,NP1</td>
<td>FTTT3930</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(12)=WORK(K)</td>
<td>FTTT3940</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(12+1)=WORK(K+1)</td>
<td>FTTT3950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>693 K=K+2</td>
<td>FTTT3960</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GO TO 698</td>
<td>FTTT3970</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>FTTT3980</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>FTTT3990</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>695 J3MAX=13+1FP2-NP1</td>
<td>FTTT4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DD 697 J3=J3,J3MAX,NP1</td>
<td>FTTT4020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J2MAX=J3+NP2=J2STP</td>
<td>FTTT4030</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DD 697 J2=J2,J2MAX,J2STP</td>
<td>FTTT4040</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J1MAX=J2*J1+2=1FP2</td>
<td>FTTT4050</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J1CNJ=J1<em>J2MAX</em>J2STP=J2</td>
<td>FTTT4060</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DD 697 J1=J1,J1MAX,1FP2</td>
<td>FTTT4070</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K=K+1,J1=13</td>
<td>FTTT4080</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(J1)=WORK(K)</td>
<td>FTTT4090</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(J1+1)=WORK(K+1)</td>
<td>FTTT4100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF(J1=J2097,H1=696)</td>
<td>FTTT4110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF(J1+1=WORK(K+1)</td>
<td>FTTT4120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>425 IF(J1+1=H1=696)</td>
<td>FTTT4130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(J1CNJ=H1CNJ+1)=K1</td>
<td>FTTT4140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA(J1CNJ+1)=WORK(K+1)</td>
<td>FTTT4150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>696 J1CNJ=J1CNJ-1FP2</td>
<td>FTTT4160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>698 CONTINUE</td>
<td>FTTT4170</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4180</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4190</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>430 IF=IF+1</td>
<td>FTTT4210</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4220</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4230</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>435 C</td>
<td>FTTT4240</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>FTTT4250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>700 GO TO (900,800,900,701),ICASE</td>
<td>FTTT4260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>701 IHALF=N</td>
<td>FTTT4270</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=N+1</td>
<td>FTTT4280</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA=TWDR/FLOAT(N)</td>
<td>FTTT4290</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>440 IF (SIGN=703,702,702</td>
<td>FTTT4300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA=THETA</td>
<td>FTTT4310</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>702 SIN=THETA</td>
<td>FTTT4320</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>703 SIN=THETA</td>
<td>FTTT4330</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSTP=W02+2*SIN+*SIN+*SIN</td>
<td>FTTT4340</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSTP=W03+*SIN+*SIN+*SIN</td>
<td>FTTT4350</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>445 W=100</td>
<td>FTTT4360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF=IF+1</td>
<td>FTTT4370</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J=J+1</td>
<td>FTTT4380</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GO TO 725</td>
<td>FTTT4390</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>710 J=J+1</td>
<td>FTTT4400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GO 720 I=I+1,NTOT,W02</td>
<td>FTTT4410</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUM=DATA(I)+DATA(J)/2.</td>
<td>FTTT4420</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUM=DATA(I)+DATA(J+1)/2.</td>
<td>FTTT4430</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B=DATA(I)-DATA(J)/2.</td>
<td>FTTT4440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>455 C=DATA(I)+DATA(J+1)/2.</td>
<td>FTTT4450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMP=WR+SUM+1=1=1</td>
<td>FTTT4460</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE FOURT 74/74 OPT=1 PMOMP

515
520
525
530

GO 840 I=IMIN,IMAX,2
DATA(I)=DATA(J)
DATA(I+1)=DATA(J+1)
840 J=J-2
850 J=JMAX
GO 860 I=IMIN,IMAX,NPO
DATA(I)=DATA(J)
DATA(I+1)=DATA(J+1)
860 J=J-NPO
C C ENO LOOP ON EACH DIMENSION
C 900 NPO=NPI
910 NP1=NP2
920 RETURN
END

CARDNR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM
496 I DATA ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
497 I DATA ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

SYMBOLIC REFERENCE MAP (R=1)
ENTRY POINTS 3 FOURT

VARIABLES SN TYPE RELOCATION 1527 DIFI REAL
0 DATA REAL ARRAY F.P. 1527 DIFI REAL
1526 DIFR REAL ARRAY 1417 I INTEGER
1416 ICASE INTEGER 1403 IDIM INTEGER
1412 IDIV INTEGER 1411 IF INTEGER
1532 IFACT INTEGER 1475 IFP INTEGER
1473 IFP1 INTEGER 1524 IDIN INTEGER
1530 IMAX INTEGER 1413 IQOUT INTEGER
1335 IPAR INTEGER 0 IFORM INTEGER
1414 IREM INTEGER 1426 IMAX INTEGER
1427 I INTEGER 1425 I INTEGER
1421 IING INTEGER 1430 I INTEGER
1521 12MAX INTEGER 1520 J MAX INTEGER
1520 J INTEGER 1505 J INTEGER
1525 JMIN INTEGER 1510 J MAX INTEGER
1529 JNLJ INTEGER 1500 JING INTEGER
1504 JMIN INTEGER 1502 J2 INTEGER
1427 JING INTEGER 1501 J2MIN INTEGER
1476 JING INTEGER 1431 J3 INTEGER
1506 JMAX INTEGER 1512 K INTEGER
1515 KMAX INTEGER 1451 KDIF INTEGER

09/21/81 12.15.51 PAGE 10
<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1450</td>
<td>KMIN</td>
<td>INTEGER</td>
<td>1507 KANG</td>
</tr>
<tr>
<td>1452</td>
<td>KSTEP</td>
<td>INTEGER</td>
<td>1436 K1</td>
</tr>
<tr>
<td>1437</td>
<td>K2</td>
<td>INTEGER</td>
<td>1453 K3</td>
</tr>
<tr>
<td>1454</td>
<td>K4</td>
<td>INTEGER</td>
<td>1443 L</td>
</tr>
<tr>
<td>1441</td>
<td>LMAX</td>
<td>INTEGER</td>
<td>1407 M</td>
</tr>
<tr>
<td>1440</td>
<td>MAX</td>
<td>INTEGER</td>
<td>1405 N</td>
</tr>
<tr>
<td>0</td>
<td>NDIM</td>
<td>INTEGER</td>
<td>F.P.</td>
</tr>
<tr>
<td>0</td>
<td>NN</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
</tr>
<tr>
<td>1434</td>
<td>NCV2T</td>
<td>INTEGER</td>
<td>1423 NREV</td>
</tr>
<tr>
<td>1422</td>
<td>NF0</td>
<td>INTEGER</td>
<td>1404 N+1</td>
</tr>
<tr>
<td>1474</td>
<td>NRP1HF</td>
<td>INTEGER</td>
<td>1406 N*2</td>
</tr>
<tr>
<td>1424</td>
<td>NRP2HF</td>
<td>INTEGER</td>
<td>1402 N0T</td>
</tr>
<tr>
<td>1440</td>
<td>NWS3</td>
<td>INTEGER</td>
<td>1517 OLD53</td>
</tr>
<tr>
<td>1516</td>
<td>OLD3R</td>
<td>REAL</td>
<td>1503 SINH</td>
</tr>
<tr>
<td>1514</td>
<td>SUMI</td>
<td>REAL</td>
<td>1513 SUMR</td>
</tr>
<tr>
<td>1433</td>
<td>TEMPI</td>
<td>REAL</td>
<td>1432 TEMPR</td>
</tr>
<tr>
<td>1442</td>
<td>THETA</td>
<td>REAL</td>
<td>1401 TAOPI</td>
</tr>
<tr>
<td>1520</td>
<td>T2J3R</td>
<td>REAL</td>
<td>1466 T2I</td>
</tr>
<tr>
<td>1465</td>
<td>T24</td>
<td>REAL</td>
<td>1470 T31</td>
</tr>
<tr>
<td>1467</td>
<td>T3R</td>
<td>REAL</td>
<td>1472 T41</td>
</tr>
<tr>
<td>1471</td>
<td>T4R</td>
<td>REAL</td>
<td>1456 U11</td>
</tr>
<tr>
<td>1455</td>
<td>U12</td>
<td>REAL</td>
<td>1460 U21</td>
</tr>
<tr>
<td>1457</td>
<td>U2R</td>
<td>REAL</td>
<td>1462 U31</td>
</tr>
<tr>
<td>1461</td>
<td>U3R</td>
<td>REAL</td>
<td>1464 U41</td>
</tr>
<tr>
<td>1463</td>
<td>U4R</td>
<td>REAL</td>
<td>1376 W1</td>
</tr>
<tr>
<td>0</td>
<td>WORK</td>
<td>REAL</td>
<td>1375 WQ</td>
</tr>
<tr>
<td>1400</td>
<td>WSTPI</td>
<td>REAL</td>
<td>1377 WSTPR</td>
</tr>
<tr>
<td>1445</td>
<td>W21</td>
<td>REAL</td>
<td>1444 W2R</td>
</tr>
<tr>
<td>1447</td>
<td>W3I</td>
<td>REAL</td>
<td>1446 W3R</td>
</tr>
</tbody>
</table>

**EXTERNALS**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ARG</th>
<th>1 LIBRARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIN</td>
<td>REAL</td>
<td>1 LIBRARY</td>
</tr>
</tbody>
</table>

**INLINE FUNCTIONS**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ARG</th>
<th>1 INTRIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXO</td>
<td>INTEGER</td>
<td>0 INTRIN</td>
</tr>
</tbody>
</table>

**STATEMENT LABELS**

<table>
<thead>
<tr>
<th>STATEMENT</th>
<th>TYPE</th>
<th>ARG</th>
<th>1 INTRIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INACTIVE</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>36</td>
<td>10</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>46</td>
<td>20</td>
<td>47</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>32</td>
<td>INACTIVE</td>
<td>61</td>
</tr>
<tr>
<td>0</td>
<td>51</td>
<td>INACTIVE</td>
<td>66</td>
</tr>
<tr>
<td>0</td>
<td>71</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>74</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>135</td>
<td>INACTIVE</td>
<td>130</td>
</tr>
<tr>
<td>0</td>
<td>120</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>167</td>
<td>140</td>
<td>0</td>
<td>145</td>
</tr>
<tr>
<td>205</td>
<td>310</td>
<td>0</td>
<td>320</td>
</tr>
<tr>
<td>0</td>
<td>310</td>
<td>INACTIVE</td>
<td>260</td>
</tr>
<tr>
<td>0</td>
<td>370</td>
<td>0</td>
<td>370</td>
</tr>
<tr>
<td>256</td>
<td>400</td>
<td>265</td>
<td>405</td>
</tr>
<tr>
<td>362</td>
<td>420</td>
<td>0</td>
<td>430</td>
</tr>
<tr>
<td>315</td>
<td>450</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>356</td>
<td>475</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>423</td>
<td>500</td>
<td>430</td>
<td>510</td>
</tr>
<tr>
<td>0</td>
<td>539</td>
<td>INACTIVE</td>
<td>0</td>
</tr>
<tr>
<td>501</td>
<td>510</td>
<td>0</td>
<td>565</td>
</tr>
<tr>
<td>STATEMENT LABELS</td>
<td>SUBROUTINE FOUNT 74/74</td>
<td>OPT=1</td>
<td>PMDMP</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>520 600</td>
<td></td>
<td>0</td>
<td>605</td>
</tr>
<tr>
<td>0 611</td>
<td>INACTIVE</td>
<td>542</td>
<td>612</td>
</tr>
<tr>
<td>0 620</td>
<td>INACTIVE</td>
<td>553</td>
<td>625</td>
</tr>
<tr>
<td>0 635</td>
<td></td>
<td>637</td>
<td>640</td>
</tr>
<tr>
<td>645 650</td>
<td></td>
<td>0</td>
<td>655</td>
</tr>
<tr>
<td>725 665</td>
<td></td>
<td>737</td>
<td>670</td>
</tr>
<tr>
<td>771 690</td>
<td></td>
<td>0</td>
<td>685</td>
</tr>
<tr>
<td>1014 633</td>
<td></td>
<td>0</td>
<td>691</td>
</tr>
<tr>
<td>0 673</td>
<td></td>
<td>1043</td>
<td>695</td>
</tr>
<tr>
<td>1075 697</td>
<td></td>
<td>1106</td>
<td>698</td>
</tr>
<tr>
<td>1131 701</td>
<td></td>
<td>0</td>
<td>702</td>
</tr>
<tr>
<td>1153 710</td>
<td></td>
<td>0</td>
<td>720</td>
</tr>
<tr>
<td>0 730</td>
<td>INACTIVE</td>
<td>0</td>
<td>731</td>
</tr>
<tr>
<td>1230 740</td>
<td></td>
<td>1235</td>
<td>745</td>
</tr>
<tr>
<td>1344 755</td>
<td></td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>1262 770</td>
<td></td>
<td>0</td>
<td>775</td>
</tr>
<tr>
<td>1275 800</td>
<td></td>
<td>0</td>
<td>805</td>
</tr>
<tr>
<td>1315 820</td>
<td></td>
<td>0</td>
<td>830</td>
</tr>
<tr>
<td>1332 650</td>
<td></td>
<td>0</td>
<td>860</td>
</tr>
<tr>
<td>0 910</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOOPS LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
<th>INSTACK</th>
<th>EXITS</th>
<th>EXITS</th>
<th>NOT INNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>2</td>
<td>DI1M</td>
<td>89</td>
<td>91</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>910</td>
<td>IDI1M</td>
<td>96</td>
<td>528</td>
<td>1034</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>80</td>
<td>J</td>
<td>159</td>
<td>161</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>160</td>
<td>I2</td>
<td>172</td>
<td>189</td>
<td>433</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>144</td>
<td>125</td>
<td>I1</td>
<td>175</td>
<td>183</td>
<td>218</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>125</td>
<td>125</td>
<td>I3</td>
<td>176</td>
<td>193</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>340</td>
<td>J1</td>
<td>201</td>
<td>210</td>
<td>205</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>213</td>
<td>340</td>
<td>J3</td>
<td>202</td>
<td>210</td>
<td>228</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>222</td>
<td>340</td>
<td>K1</td>
<td>203</td>
<td>210</td>
<td>108</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>207</td>
<td>570</td>
<td>L</td>
<td>222</td>
<td>237</td>
<td>2215</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>303</td>
<td>530</td>
<td>J2</td>
<td>229</td>
<td>284</td>
<td>1698</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>353</td>
<td>530</td>
<td>J3</td>
<td>230</td>
<td>284</td>
<td>1698</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>332</td>
<td>520</td>
<td>K1</td>
<td>236</td>
<td>279</td>
<td>1228</td>
<td></td>
<td></td>
<td>OPT NOT INNER</td>
</tr>
<tr>
<td>547</td>
<td>635</td>
<td>J2</td>
<td>319</td>
<td>342</td>
<td>703</td>
<td></td>
<td></td>
<td>OPT NOT INNER</td>
</tr>
<tr>
<td>576</td>
<td>635</td>
<td>J1</td>
<td>329</td>
<td>342</td>
<td>438</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>575</td>
<td>630</td>
<td>I1</td>
<td>331</td>
<td>339</td>
<td>255</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>577</td>
<td>630</td>
<td>I3</td>
<td>334</td>
<td>339</td>
<td>21B</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>606</td>
<td>630</td>
<td>J1</td>
<td>336</td>
<td>339</td>
<td>63</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>664</td>
<td>633</td>
<td>I1</td>
<td>351</td>
<td>428</td>
<td>239J</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>663</td>
<td>633</td>
<td>I3</td>
<td>352</td>
<td>428</td>
<td>2208</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>664</td>
<td>633</td>
<td>KMIN</td>
<td>353</td>
<td>400</td>
<td>1348</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>670</td>
<td>649</td>
<td>J1</td>
<td>355</td>
<td>392</td>
<td>1074</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>674</td>
<td>640</td>
<td>J3</td>
<td>357</td>
<td>392</td>
<td>1069</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>715</td>
<td>610</td>
<td>J2</td>
<td>336</td>
<td>365</td>
<td>25</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>1075</td>
<td>692</td>
<td>I2</td>
<td>405</td>
<td>400</td>
<td>45</td>
<td></td>
<td></td>
<td>INSTACK</td>
</tr>
<tr>
<td>1047</td>
<td>617</td>
<td>J3</td>
<td>415</td>
<td>427</td>
<td>375</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>1051</td>
<td>634</td>
<td>J1</td>
<td>417</td>
<td>427</td>
<td>358</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>1062</td>
<td>677</td>
<td>J1</td>
<td>420</td>
<td>427</td>
<td>103</td>
<td></td>
<td></td>
<td>OPT NOT INNER</td>
</tr>
<tr>
<td>1112</td>
<td>720</td>
<td>I</td>
<td>451</td>
<td>462</td>
<td>178</td>
<td></td>
<td></td>
<td>OPT NOT INNER</td>
</tr>
<tr>
<td>1275</td>
<td>715</td>
<td>I</td>
<td>470</td>
<td>471</td>
<td>25</td>
<td></td>
<td></td>
<td>INSTACK</td>
</tr>
<tr>
<td>1301</td>
<td>810</td>
<td>I1</td>
<td>504</td>
<td>522</td>
<td>524</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>1302</td>
<td>810</td>
<td>I2</td>
<td>506</td>
<td>522</td>
<td>4708</td>
<td></td>
<td></td>
<td>NOT INNER</td>
</tr>
<tr>
<td>1126</td>
<td>840</td>
<td>I</td>
<td>514</td>
<td>517</td>
<td>38</td>
<td></td>
<td></td>
<td>INSTACK</td>
</tr>
<tr>
<td>1331</td>
<td>850</td>
<td>I</td>
<td>519</td>
<td>522</td>
<td>33</td>
<td></td>
<td></td>
<td>INSTACK</td>
</tr>
</tbody>
</table>
SUBROUTINE GAUSS

SUBROUTINE GAUSS(I,A,Y,VAL)
THIS ROUTINE CALCULATES A GAUSSIAN DISTRIBUTED RANDOM VARIABLE
VAL, WITH MEAN=0. AND STANDARD DEVIATION=1.
IA IS INITIALIZED BEFORE FIRST CALL TO ANY ODD INTEGER LESS THAN
10 DIGITS IN LENGTH.
IY IS GENERATED AND SHOULD BE USED FOR IA ON THE NEXT CALL TO
THIS ROUTINE.
VAL=0.
DO 1 I=1,12
X=RANF(DUM)
CALL RANDOM(IA,IY,X)
IA=IY
VAL=VAL+X
1 CONTINUE
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
  3 GAUSS

VARIABLES  SN  TYPE   RELOCATION
          0   REAL   *UNDEF  24  I   INTEGER
          0   INTEGER  F.P.  0  IY  INTEGER  F.P.
          0   REAL  F.P.

INLINE FUNCTIONS  TYPE  ARGS
RANF  REAL  1  INTRIN

STATEMENT LABELS
  0  I

LOOPS  LABEL  INDEX  FROM-TO  LENGTH  PROPERTIES
  12  I   10 15   68  INSTACK

STATISTICS
  PROGRAM LENGTH   338  CM USED
  520008 CM USED
SUBROUTINE RANDOM

1

C SUBROUTINE RANDOM(IX,IY,YFL)
C THIS ROUTINE GENERATES A UNIFORMLY DISTRIBUTED RANDOM VARIABLE, VAL,
C WITH VALUE BETWEEN 0. AND 1.
C
5
C IA IS SHOULD BE INITIALIZED BEFORE THE FIRST CALL TO THIS ROUTINE
C TO AN ODD INTEGER LESS THAN 10 DIGITS IN LENGTH.
C IY IS GENERATED BY THE PROGRAM AND SHOULD BE USED FOR IA ON THE NEXT
C CALL TO THIS ROUTINE.
C
10
C IY=IX*C5539
C IF(IY) 5,6,6
C 5 IY=IY+2147483647+1
C 6 YFL=IY
C YFL=YFL*.4656613E-9
C RETURN
C
15
C END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 RANDOM

VARIABLES SN  TYPE     LOCATION  SN  TYPE     LOCATION
0  IX  INTEGER     F.P.  0  IY  INTEGER     F.P.
0  YFL REAL        F.P.

STATEMENTS LABELS
0 5 INACTIVE 12 6

STATISTICS
PROGRAM LENGTH 20B 16
52000B CM USED
SUBROUTINE MULT 74/74 OPT=1 PMOHP

1
SUBROUTINE MULT(A,B,L,M,N,C)
REAL A(L,M),B(M,N),C(L,N)
DO 300 I=1,L
DO 200 J=1,N
C(I,J)=0.
DO 100 INDEX=1,M
C(I,J)=C(I,J)+A(I,INDEX)*B(INDEX,J)
100 CONTINUE
200 CONTINUE
10
300 CONTINUE
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 MULT

VARIABLES SN TYPE RELOCATION
0 A REAL ARRAY F.P.
0 C REAL ARRAY F.P.
46 INDEX INTEGER
0 L INTEGER F.P.
0 N INTEGER F.P.

STATEMENT LABELS
0 100 0 200 0 300

LOOPS LABEL INDEX FROM-TO LENGTH PROPERTIES
12 300 I 3 10 27B NOT INNER
13 200 J 4 9 24B NOT INNER
30 100 INDEX 6 8 39 INSTACK

STATISTICS
PROGRAM LENGTH 578
52000B CM USED
SUBROUTINE SPTN(SIGMA,B,R,M)
DIMENSION R(64,64),C(5)
DATA C/3679.,2431.,1353.,1069.,0591/

5      C
       SET UP SPATIAL NOISE CORRELATION COEFFICIENT MATRIX
       USING SECOND NEAREST NEIGHBOR CORRELATION.
       C IS THE ARRAY CONTAINING THE NON-ZERO ELEMENTS
       CORRESPONDING TO THE DISTANCES TO NEIGHBORING PIXELS.
       THE ARRAY VALUES ARE EXP(-DISTANCE IN PIXELS)
       SIGMA IS THE BACKGROUND VARIANCE
       R IS THE M**2 BY R**2 CORRELATION MATRIX

10      C
     N=M**2
     DD 30 I=1,N
     DD 30 J=1,N
     R(I,J)=0.0
     CONTINUE

20      DD 36 I=1,N
     R(I,1)=1.
     IF (I.GE.64) GO TO 36
     R(I,1+1)=C(1)
     IF (I.GE.63) GO TO 36
     R(I,1+2)=C(3)
     IF (I.GE.59) GO TO 36
     R(I,1+6)=C(14)
     IF (I.GE.50) GO TO 35
     R(I,1+7)=C(2)
     IF (I.GE.57) GO TO 36
     R(I,1+9)=C(1)
     IF (I.GE.56) GO TO 36
     R(I,1+9)=C(2)
     IF (I.GE.55) GO TO 36
     R(I,1+10)=C(4)
     IF (I.GE.51) GO TO 36
     R(I,1+14)=C(5)
     IF (I.GE.50) GO TO 36
     R(I,1+15)=C(4)
     IF (I.GE.49) GO TO 36
     R(I,1+16)=C(3)
     IF (I.GE.48) GO TO 36
     R(I,1+17)=C(4)
     IF (I.GE.47) GO TO 36
     R(I,1+18)=C(5)
     CONTINUE

30      DD 37 I=1,M
     R(I+1,1)=0.0
     R(I+1,7,1)=0.0
     R(I+1,7,2+1)=0.0
     R(I+1,5,1)=0.0
     IF (I.GE.3) GO TO 37
     R(I+1,4+1)=0.0
     R(I+1,6+1)=0.0
     R(I+1,8+1)=0.0
     R(I+1,7,8+1)=0.0
     R(I+1,7,8+1+1)=0.0

35
SUBROUTINE SPTN  74/74  OPT=1  PMDMP

R(8*1-6.8*(+8))=0.0
IF (I.GE.7) GO TO 37
R(8*1-6.8*(+8))=0.0
R(8*1.8*1+10)=0.0
R(8*1.8*1+9)=0.0
IF (I.GE.6) GO TO 37
R(8*1.8*1+7)=0.0
R(8*1.8*1+18)=0.0
R(8*1.8*1+17)=0.0
CONTINUE
DO 38 I=1,N
L=I+1
DO 38 J=L,N
IF(L.GT.N) GO TO 38
R(J,I)=R(I,J)
CONTINUE
DO 39 I=1,N
DO 39 J=1,N
R(I,J)=SIGMAB*R(I,J)
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 SPTN

VARIABLES SN TYPE ACTION RELOCATION
163 C REAL ARRAY 0 I 160 INTEGER
151 J INTEGER 162 L INTEGER
0 M INTEGER 157 N INTEGER
0 R REAL ARRAY 0 SIGMAB REAL

STATEMENT LABELS
0 30 166 36
133 38 113 37

LOOP INDEX FROM TO LENGTH PROPERTIES
11 30 I 17 20 133 NOT INNER
16 30 J 18 20 28 INSTACK
17 30 I 21 47 303 OPT
18 37 I 48 67 172 OPT
117 36 I 68 73 218 NOT INNER
131 39 J 70 73 48 INSTACK
131 39 I 74 76 148 NOT INNER
145 39 J 75 76 30 INSTACK

STATISTICS
PROGRAM LENGTH 176B 126
SUBROUTINE INVERT

1
SUBROUTINE INVERT(A,N,B,IER)
IMPLICIT REAL (A-H,O-Z)
DIMENSION A(1),B(1)
REAL L(128),M(128)
5
NSQ=N*N
DO 1000 I=1,NSQ
1000
B(I)=A(I)
D=1.0
NK=N
10
DO 50 K=1,N
NK=NK+K
L(K)=K
M(K)=K
KK=NK+K
15
BIGA=A(KK)
DO 20 J=K,N
IZ=N+(J-1)
DO 20 I=K,N
IJ=12*1
20
IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15
BIGA=A(IJ)
L(K)=I
M(K)=J
25
CONTINUE
J=K
IF(J-K) 35,35,25
25
KI=K-N
DO 30 I=1,N
KI=KI+N
30
HOLD=A(KI)
JI=KI-K+J
A(KJ)=A(JI)
30
A(JJ)=HOLD
35
I=MI(K)
IF(I-K) 45,45,38
38
JP=N*(I-1)
DO 40 J=1,N
JK=NK+J
JI=JP+J
40
HOLD=A(JK)
A(JK)=A(JI)
A(JJ)=HOLD
45
IF(BIGA) 48,46,48
46
D=0.0
45
IER=129
GO TO 150
48
DO 55 I=1,N
IF(I-K) 50,55,50
50
IK=K+I
55
A(IK)=A(IK)/(-BIGA)
55
CONTINUE
65
IZ=1,N
65
I=I-N
DO 65 J=1,N
LJ=J+N
IF(I-K) 60,65,60
60
END
SUBROUTINE INVERT  74/74  OPT=1  PMOMP  FTN 4.8+528  09/21/81  12.15.51  PAGE 2

IF(J-K) 62,65,62
KJ=J-I+K
A(IJ)=A(IK)+A(KJ)+A(IJ)
    CONTINUE
    KJ=K-N
    DO 75 J=1,N
    KJ=KJ+N
65 IF(J-K) 70,75,70
    A(KJ)=A(KJ)/BIGA
    CONTINUE
    D=0BIGA
    A(KK)=1.0/BIGA
70 CONTINUE
    K=N
    K=(K-1)
75 IF(K) 150,150,105
    I=I(K)
70 IF(I-K) 120,120,108
    JQ=N*(K-1)
75 IF(J-K) 120,120,108
    JR=N*(I-1)
    DO 110 J=1,N
    JK=JQ+J
    HOLD=A(JK)
    JI=JR+J
    A(JK)=A(JI)
    A(JI)=HOLD
110 J=M(K)
120 IF(J-K) 100,100,125
125 K1=K-N
    DO 130 I=1,N
    K1=K1+N
    HOLD=A(K1)
    J1=KI-K+J
    A(K1)=A(J1)
130 A(J1)=HOLD
    GO TO 100
150 DO 1002 I=1,NSQ
70 SAVE=A(I)
    A(I)=B(I)
    B(I)=SAVE
1002 CONTINUE
    RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
3 INVERT

VARIABLES   SN   TYPE   RELOCATION
0   A   REAL   ARRAY   F.P.
306   BIGA   REAL
313   HOLD   REAL

0   B   REAL   ARRAY   F.P.
302   D   REAL
301   I   INTEGER
<table>
<thead>
<tr>
<th>SUBROUTINE INTOVERT</th>
<th>74/74</th>
<th>OPT=1</th>
<th>PMDMP</th>
<th>FTN 4.8+52B</th>
<th>09/21/81</th>
<th>12.15.81</th>
<th>PAGE</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARIABLES SN TYPE</td>
<td></td>
<td></td>
<td></td>
<td>RELOCATION</td>
<td>F.P.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 IER INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>311 IJ</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>317 IK INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>310 IZ</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>327 J INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>314 J1</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>316 JK INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>315 JP</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>321 JO INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>322 JR</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>304 K INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>312 K1</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>320 KJ INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>305 KK</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>322 L REAL ARRAY</td>
<td></td>
<td></td>
<td></td>
<td>303 NK</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300 NSQ INTEGER</td>
<td></td>
<td></td>
<td></td>
<td>323 SAVE</td>
<td>REAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARS REAL ARRAY</td>
<td></td>
<td></td>
<td></td>
<td>524 M</td>
<td>REAL</td>
<td>524 M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INLINE FUNCTIONS TYPE ARGS</td>
<td></td>
<td></td>
<td></td>
<td>ABS REAL 1 INTRIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATEMENT LABELS</td>
<td></td>
<td></td>
<td></td>
<td>53 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 10 INACTIVE</td>
<td>0 15</td>
<td>0 15</td>
<td>0 15</td>
<td>INACTIVE 53 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 25 INACTIVE</td>
<td>0 30</td>
<td>0 30</td>
<td>0 30</td>
<td>0 30 100 35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 38 INACTIVE</td>
<td>0 40</td>
<td>0 40</td>
<td>0 40</td>
<td>0 40 121 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 46 INACTIVE</td>
<td>125 48</td>
<td>125 48</td>
<td>125 48</td>
<td>0 46 0 50</td>
<td>INACTIVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>137 55 INACTIVE</td>
<td>0 60</td>
<td>0 60</td>
<td>0 60</td>
<td>0 60 0 62</td>
<td>INACTIVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165 65 INACTIVE</td>
<td>0 70</td>
<td>0 70</td>
<td>0 70</td>
<td>0 70 203 75</td>
<td>INACTIVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 80 INACTIVE</td>
<td>215 100</td>
<td>215 100</td>
<td>215 100</td>
<td>0 80 0 105</td>
<td>INACTIVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 10B INACTIVE</td>
<td>0 110</td>
<td>0 110</td>
<td>0 110</td>
<td>0 110 241 120</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 125 INACTIVE</td>
<td>0 130</td>
<td>0 130</td>
<td>0 130</td>
<td>0 130 263 150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1000 INACTIVE</td>
<td>0 1002</td>
<td>0 1002</td>
<td>0 1002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STATISTICS

PROGRAM LENGTH 7568 494

520008 CM USED
SUBROUTINE CHOLY

C THIS ROUTINE DETERMINES THE LOWER TRIANGULAR CHOLESKY SQUARE-
C ROOT OF AN N X N MATRIX.
C A IS THE INPUT MATRIX AND S IS THE CHOLESKY SQUARE-ROOT MATRIX.

DIMENSION A(N,N),S(N,N)
DO 1 I=1,N
    DO 1 J=1,N
        S(I,J)=0.
    1 CONTINUE
DO 123 I=1,N
    DO 123 J=1,N
        IF(A(I,J)).GT.1.E-06) GO TO 124
    123 CONTINUE
RETURN
CONTINUE
124 S(I,1)=SQRT(A(1,1))
DO 5 I=2,N
    IM1=I-1
    DO 5 J=1,IM1
        JM1=J-1
        SUM=0.
        DO 2 K=1,IM1
            SUM=SUM+S(I,K)*S(J,K)
    2 SUM=SUM+S(I,J)**2
        S(I,J)=(A(I,J)-SUM)/S(J,J)
4 SUM=SUM+S(I,K)**2
5 S(I,1)=SQRT(A(I,1)-SUM)
RETURN
END

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
  3 CHOLY

VARIABLES SN TYPE  RELOCATION
  0 A  REAL ARRAY F.P.  121 I  INTEGER
  123 IM1  INTEGER  122 J  INTEGER
  124 JM1  INTEGER  126 K  INTEGER
  0 N  INTEGER F.P.  0 S  REAL ARRAY F.P.
  125 SUM  REAL

EXTERNALS  TYPE  ARGS
  SQRT  REAL  1 LIBRARY

INLINE FUNCTIONS  TYPE  ARGS
  ABS  REAL  1 INTRIN

STATEMENT LABELS
  0 1  0 2  0 3
  0 4  0 5  0 123
  40 124
<table>
<thead>
<tr>
<th>SUBROUTINE CHOLY</th>
<th>OPT+1</th>
<th>PMDMP</th>
<th>74/74</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOOPS</td>
<td>LABEL</td>
<td>INDEX</td>
<td>FROM-TO</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>I</td>
<td>6 8</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>J</td>
<td>7 8</td>
</tr>
<tr>
<td>24</td>
<td>123</td>
<td>I</td>
<td>9 12</td>
</tr>
<tr>
<td>25</td>
<td>123</td>
<td>J</td>
<td>10 12</td>
</tr>
<tr>
<td>43</td>
<td>5</td>
<td>I</td>
<td>16 27</td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>J</td>
<td>18 23</td>
</tr>
<tr>
<td>57</td>
<td>2</td>
<td>K</td>
<td>21 22</td>
</tr>
<tr>
<td>102</td>
<td>4</td>
<td>K</td>
<td>25 26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATISTICS</th>
<th>PROG LENGTH</th>
<th>146B</th>
<th>102</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM LENGTH</td>
<td>146B</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>520000 CM USED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAUSSIAN TARGET COVARIANCE VALUE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF ZENOEUS TO PAD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF FRAMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF SIMULATIONS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALPHA FOR SMOOTHING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF HIGH FREQ COMPONENTS TO ZERO</td>
<td>40000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPUT BACKGROUND VARIANCE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIANCE OF TRUTH MODEL DYNAMICS UNCERTAINTY</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAME XERR(-)</th>
<th>SXERR(+)</th>
<th>YERR(-)</th>
<th>SYERR(+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.</td>
<td>0.0.</td>
<td>0.0.</td>
<td>0.0.</td>
</tr>
<tr>
<td>2 -0.14272E-01</td>
<td>0.37929E-01</td>
<td>-0.41427E-01</td>
<td>0.84031E-01</td>
</tr>
<tr>
<td>3 -0.25117E-01</td>
<td>0.75213E-01</td>
<td>-0.50213E-01</td>
<td>0.85666E-01</td>
</tr>
<tr>
<td>4 -0.20414E-01</td>
<td>0.64372E-01</td>
<td>-0.93329E-01</td>
<td>0.91635E-01</td>
</tr>
<tr>
<td>5 -0.17489E-01</td>
<td>0.86850E-01</td>
<td>-0.66836E-01</td>
<td>0.96284E-01</td>
</tr>
<tr>
<td>6 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.60376E-01</td>
<td>0.82147E-01</td>
</tr>
<tr>
<td>7 -0.14147E-01</td>
<td>0.91384E-01</td>
<td>-0.10172E+00</td>
<td>0.91925E-01</td>
</tr>
<tr>
<td>8 -0.13538E-01</td>
<td>0.83448E-01</td>
<td>-0.13201E+00</td>
<td>0.10202E+00</td>
</tr>
<tr>
<td>9 -0.12136E-01</td>
<td>0.78448E-01</td>
<td>-0.12032E+00</td>
<td>0.87223E-01</td>
</tr>
<tr>
<td>10 -0.11350E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>11 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>12 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>13 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>14 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>15 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>16 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>17 -0.11398E-01</td>
<td>0.99137E-01</td>
<td>-0.16113E+00</td>
<td>0.92622E-01</td>
</tr>
<tr>
<td>FRAME</td>
<td>CNR(-)</td>
<td>SCN(-)</td>
<td>YCER(-)</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>-1.6982E-01</td>
<td>.55503E-01</td>
<td>-1.7432E-01</td>
</tr>
<tr>
<td>3</td>
<td>-1.7432E-01</td>
<td>.67609E-01</td>
<td>-1.4007E+00</td>
</tr>
<tr>
<td>4</td>
<td>-1.1673E+00</td>
<td>.75470E-01</td>
<td>-1.2902E+00</td>
</tr>
<tr>
<td>5</td>
<td>-1.9165E-01</td>
<td>.62306E-01</td>
<td>-2.0440E+00</td>
</tr>
<tr>
<td>6</td>
<td>-1.0002E+00</td>
<td>.10859E+00</td>
<td>-1.6278E+00</td>
</tr>
<tr>
<td>7</td>
<td>-1.0903E+00</td>
<td>.10942E+00</td>
<td>-1.9075E+00</td>
</tr>
<tr>
<td>8</td>
<td>-1.0409E+00</td>
<td>.10470E+00</td>
<td>-1.5933E+00</td>
</tr>
<tr>
<td>9</td>
<td>-1.0948E+00</td>
<td>.10578E+00</td>
<td>-1.5591E+00</td>
</tr>
<tr>
<td>10</td>
<td>-8.4578E+00</td>
<td>.06344E-01</td>
<td>-1.9815E+00</td>
</tr>
<tr>
<td>11</td>
<td>-1.2967E+00</td>
<td>.76120E-01</td>
<td>-1.6110E+00</td>
</tr>
<tr>
<td>12</td>
<td>-1.0494E+00</td>
<td>.99747E-01</td>
<td>-2.1675E+00</td>
</tr>
<tr>
<td>13</td>
<td>-1.4920E+00</td>
<td>.98517E-01</td>
<td>-1.9647E+00</td>
</tr>
<tr>
<td>14</td>
<td>-1.3257E+00</td>
<td>.11088E+00</td>
<td>-2.2165E+00</td>
</tr>
<tr>
<td>15</td>
<td>-1.7629E+00</td>
<td>.10401E+00</td>
<td>-2.1630E+00</td>
</tr>
<tr>
<td>16</td>
<td>-1.5997E+00</td>
<td>.91846E-01</td>
<td>-2.4017E+00</td>
</tr>
<tr>
<td>17</td>
<td>-1.2640E+00</td>
<td>.79303E-01</td>
<td>-2.2507E+00</td>
</tr>
<tr>
<td>18</td>
<td>-1.6032E+00</td>
<td>.80530E-01</td>
<td>-2.2101E+00</td>
</tr>
<tr>
<td>19</td>
<td>-1.1067E+00</td>
<td>.64383E-01</td>
<td>-2.4307E+00</td>
</tr>
<tr>
<td>20</td>
<td>-1.1256E+00</td>
<td>.10033E+00</td>
<td>-2.2891E+00</td>
</tr>
</tbody>
</table>
### Gaussian Covariance

**Rounds: 20**

**Gaussian Covariance: 2.00**

**Background Covariance: 1.0**

**Number Zero Pad: 8**

**Smoothing Alpha: .100**

**Truth Model Uncertainty: .10**

<table>
<thead>
<tr>
<th>FRAME</th>
<th>MEAN ERROR OF ERROR IN H</th>
<th>MEAN VARIANCE OF ERROR IN D/OY</th>
<th>MEAN ERROR OF ERROR IN D/OY</th>
<th>MEAN VARIANCE OF ERROR IN D/OY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-.63852E-01</td>
<td>.77407E+00</td>
<td>.29364E+00</td>
<td>.91872E+00</td>
</tr>
<tr>
<td>2</td>
<td>-.78134E-01</td>
<td>.65121E+00</td>
<td>.20909E+00</td>
<td>.52321E+00</td>
</tr>
<tr>
<td>3</td>
<td>-.65521E-01</td>
<td>.31401E+00</td>
<td>.28246E+00</td>
<td>.35680E+00</td>
</tr>
<tr>
<td>4</td>
<td>-.65097E-01</td>
<td>.25642E+00</td>
<td>.25626E+00</td>
<td>.29197E+00</td>
</tr>
<tr>
<td>5</td>
<td>-.66417E-01</td>
<td>.22019E+00</td>
<td>.25812E+00</td>
<td>.25664E+00</td>
</tr>
<tr>
<td>6</td>
<td>-.62624E-01</td>
<td>.20263E+00</td>
<td>.24026E+00</td>
<td>.22168E+00</td>
</tr>
<tr>
<td>7</td>
<td>-.63108E-01</td>
<td>.17134E+00</td>
<td>.25264E+00</td>
<td>.19121E+00</td>
</tr>
<tr>
<td>8</td>
<td>-.63673E-01</td>
<td>.14960E+00</td>
<td>.25402E+00</td>
<td>.16760E+00</td>
</tr>
<tr>
<td>9</td>
<td>-.60624E-01</td>
<td>.13632E+00</td>
<td>.25443E+00</td>
<td>.15475E+00</td>
</tr>
<tr>
<td>10</td>
<td>-.44639E-01</td>
<td>.12053E+00</td>
<td>.26013E+00</td>
<td>.14437E+00</td>
</tr>
<tr>
<td>11</td>
<td>-.47623E-01</td>
<td>.12221E+00</td>
<td>.26158E+00</td>
<td>.13393E+00</td>
</tr>
<tr>
<td>12</td>
<td>-.44029E-01</td>
<td>.11590E+00</td>
<td>.26573E+00</td>
<td>.12471E+00</td>
</tr>
<tr>
<td>13</td>
<td>-.43711E-01</td>
<td>.11930E+00</td>
<td>.27122E+00</td>
<td>.11574E+00</td>
</tr>
<tr>
<td>14</td>
<td>-.44453E-01</td>
<td>.10519E+00</td>
<td>.26560E+00</td>
<td>.10832E+00</td>
</tr>
<tr>
<td>15</td>
<td>-.50219E-01</td>
<td>.10563E+00</td>
<td>.27118E+00</td>
<td>.10359E+00</td>
</tr>
<tr>
<td>16</td>
<td>-.46306E-01</td>
<td>.97677E-01</td>
<td>.27219E+00</td>
<td>.97111E-01</td>
</tr>
<tr>
<td>17</td>
<td>-.49134E-01</td>
<td>.97255E-01</td>
<td>.27263E+00</td>
<td>.91025E-01</td>
</tr>
<tr>
<td>18</td>
<td>-.55025E-01</td>
<td>.95349E-01</td>
<td>.28101E+00</td>
<td>.89119E-01</td>
</tr>
<tr>
<td>19</td>
<td>-.63420E-01</td>
<td>.93616E-01</td>
<td>.28339E+00</td>
<td>.86318E-01</td>
</tr>
<tr>
<td>20</td>
<td>-.66705E-01</td>
<td>.92066E-01</td>
<td>.27789E+00</td>
<td>.84924E-01</td>
</tr>
</tbody>
</table>

**Gaussian Target Covariance Value**
Correlator-Kalman Filter
Modcomp Software
DATA STRUCTURES TO CATCH STATISTICS ON FILTER

TRACKER CAPABILITY

XREF IS THE ERROR BETWEEN THE PREDICTED A DYNAMIC LOCATION AT A PARTICULAR TIME AND THE TRUTH MODEL TRUE

XREF IS THE SQUARE OF THE XREF

NOTE THAT XREF AND XREF2 ARE ARRAYS WHICH ARE DIMENSIONED TO

IF RAVE THE FIRST ROW IN EACH IS USED FOR THE X DIRECTION WHILE THE SECOND ROW IS FOR THE Y DIRECTION

YREF IS THE ERROR IN THE PREDICTED LOCATION OF THE CENTER OF AT

A PARTICULAR TIME COMPARED TO THE TRUTH MODEL

YREF2 IS THE SQUARE OF YREF

NOTE AGAIN THE DIMENSION OF XREF AND XREF2 ARE

REAL XREF(2*20),XFPL(2*20),XFPL(2*20),XFPE(2*20)

XREF IS THE ERROR BETWEEN THE UPDATED DYNAMIC LOCATION AT A PARTICULAR TIME AND THE TRUTH MODEL TRUE DYNAMIC

XFPE IS THE SQUARE OF XFPE

NOTE THE DIMENSIONALITY OF XPE,XFPE FOR THE SAME REASONS AS

ABOVE THE 20 ARE THE COMPLETION OF STATISTICS PER

FRAME OF ITS

CMF IS THE CENTROID ERROR AT THE PLUS TIME

CMFIC

REAL XPE(2*20),XPE(2*20),XPE(2*20),XPE(2*20)

FILTERS DATA STRUCTURES

PHIF IS THE STATE TRANSITION MATRIX FOR THE KALMAN FILTER

SEE SUBROUTINE PHIF

GFD IF THE RESULT OF THE INTEGRAL TEMP IN THE PROPAGATION

OF THE COV MATRIX SEE SUBROUTINE PFD

FEP IS THE FILTER CORRECTION MATRIX PLUS AFTER INTEGRATION

OF A MEASUREMENT

PHIF IS THE FILTER CORRECTION MATRIX MINUS AFTER PROPAGATION

PHIF IS THE FILTER STATE VECTOR PLUS

PHIF IS THE FILTER STATE VECTOR MINUS

REAL PHIF(4*4),CFG(4*4),PFD(4*4),PFD(4*4),XEP(4*4),XEP(4*4)
C CLASSIC FORM IV 01 09-29-61 09:13 MODIFIL PAGE 2

66 C IN THE KALMAN FILTER MEASUREMENT VECTOR
67 C REAL 7 (64)
69 C DATA 1.0/2.0/2.4/1.
72 C INITIALIZE THE FILTERS DATA STRUCTURES
73 C DATA 0.0/0.0/0.0/0.0
74 C DATA 1.0/1.0/1.0/1.0
75 C IF (P, &/0.0/0.0)
76 C IF (P, C/0.0/0.0)
77 C IF (P, &/0.0/0.0)
78 C IF (P, C/0.0/0.0)
79 C IF (P, &/0.0/0.0)
80 C IF (P, C/0.0/0.0)
81 C IF (P, &/0.0/0.0)
82 C IF (P, C/0.0/0.0)
83 C IF (P, &/0.0/0.0)
84 C IF (P, C/0.0/0.0)
85 C IF (P, &/0.0/0.0)
86 C IF (P, C/0.0/0.0)
87 C IF (P, &/0.0/0.0)
88 C IF (P, C/0.0/0.0)
89 C IF (P, &/0.0/0.0)
90 C IF (P, C/0.0/0.0)
91 C IF (P, &/0.0/0.0)
92 C IF (P, C/0.0/0.0)
93 C IF (P, &/0.0/0.0)
94 C IF (P, C/0.0/0.0)
95 C IF (P, &/0.0/0.0)
96 C IF (P, C/0.0/0.0)
97 C IF (P, &/0.0/0.0)
98 C IF (P, C/0.0/0.0)
99 C IF (P, &/0.0/0.0)
100 C IF (P, C/0.0/0.0)
101 C IF (P, &/0.0/0.0)
102 C IF (P, C/0.0/0.0)
103 C IF (P, &/0.0/0.0)
104 C IF (P, C/0.0/0.0)
105 C IF (P, &/0.0/0.0)
106 C IF (P, C/0.0/0.0)
107 C IF (P, &/0.0/0.0)
108 C IF (P, C/0.0/0.0)
109 C IF (P, &/0.0/0.0)
110 C IF (P, C/0.0/0.0)
CLASSIC FORMAT IV  DATE  DO-24 L I  09:13  I NGCFILE PAGE 3

111   SIG(1)=G.
112   SIG(2):=G(1)
113   SIG(3):=G(1)
114   SIG(4):=G(1)
115   SIG(5):=G(1)
116
117   C  INITIALIZE TARGET INTENSITY ASSUMING
118   C  A CIRCULAR CROSS-SECTION GAUSSIAN TARGET.
119   C
120   MAX(1):=20.
121   TMAX(1)=100.
122   TMAX(2)=500.
123   C  DEFINE TRUTH MODEL DYNAMICS
124   C
125   R(M) = [M]*.5
126   C  FILTER SETUP: DEV OF TRUTH MODEL ATTRIBETAL JITTER
127   C  +F0(*1.0) SIGOT
128   C  CALL TRUTH(MIT=1,CHOT=1,SIGOT=OT)
129   C
130   C  INITIALIZE THE FILTERS PARAMETERS
131   C
132   C  INITIALIZE THE FILTERS MATRICES DEFINITION
133   C
134   C  CALL FILTER(TDF=VARVF1,TF=VARVF2,TUP=PHIF+QOF)
135   C
136   C  INITIALIZE THE FILTER LARGE MATRICES TO ZERO
137   C
138   DD 21 1=120.
139   DD 21 J=1,20
140   X(W)=(.1.0)
141   Y(W)=0.
142   C(W)=0.
143   C(W)=0.
144   C(W)=0.
145   X(W)=0.
146   Y(W)=0.
147   C(W)=0.
148   DD 21 J=1,20.
149   C
150   C  USING FIRST AND SECOND NEAREST NEIGHBOR DETERMINE THE CHOLESKY
151   C  SQUAREROOT OF THE MEASURED COVARIANCE MATRIX R.
152   C
153   C  CALL SPCHF(VNP=H+K)
154   C  THIS LOOP MAKES SPATIALLY CORRELATED/UNCORRELATED NOISE
155   C  COVARIANCE THE NEXT FOUR LINES IF WANT SPATIAL CORRELATION
156   C  LD E=1.04
157   C  J=1.04
158   C  K(1,J)=G
159   C  IF(A(J)*G)
160   C  CONTINUE
161   C  PREVIOUS VERSION OF CHOLESKY PUTS ROOT BACK INTO CALLING MATRIX
162   C  CALL CHOLESKY(INP=H+K)
163   C
164   C  FINISH INITIALIZATION
165   C

P.S. The text appears to be a fragment of a FORTRAN program, possibly related to signal processing or data analysis.
**VEE1: MONTE CARLO SIMULATION**

MAKE RANDOM SIMULATIONS OF AFRAMES EACH FOR MONTE-CARLO ANALYSIS

CO 60 N=1, NFRAMES

C INITIALIZE SMOOTHED DATA ARRAY
CO 7 I=1,4
CO 7 J=1,4
I DATA(I,J) = CMPLX(CO,GO+)

C INITIALIZE STATE VECTOR
CO 71 I=1,4
XT(I,1) = 0.0
YT(I,1) = 0.0
XT(I,2) = 0.0
YT(I,2) = 0.0

C SPEC INITIAL CONDITION'S FOR THE FILTER
CO 17 E = 1.4
CO 166 = 1.4
FFK(I,J) = 0.0
WF(I,J) = 0.0
WF(I,J) = 0.0
WFF(I,J) = 0.0
WFF(I,J) = 0.0

C INITIAL CONDITIONS ON DYNAMIC STATES
YFP(I,J) = 0.0
YFP(I,J) = 0.0
YFP(I,J) = 0.0
YFP(I,J) = 0.0

C LEFT CORNER OF FIELD OF VIEW
OFFILE UPPER-LEFT CORNER OF FOV

X = YFP(I,J) - 4
Y = YFP(I,J) - 4

C NEXT TARGET FOR AFRAME FRAMES (TIME SLICES)
DD = G N=1, NFRAMES

OFFLINE GAUSSIAN PEAK LOCATIONS BASED ON CF-TOID POSITION, YT
221 YMA(1)=YT(1,1)
222 YMA(2)=YT(1,1)+2
223 YMA(3)=YT(1,1)+2
224 YMA(4)=YT(1,1)+2
225 YMA(5)=YT(2,1)+1
226 YMA(6)=YT(2,1)+1
227 C GET MEASUREMENT NOISE ARRAY
228 C CALL NOISE(XX,64)
229 C CALL MULT(DA4,X+4,1,64)
230 C GET MEASUREMENT DATA
231 C CALL IDEALL(YMAX,K,XMAX,YMAX+2*K,ZX,ZY,DATA*DX*DY)
232 C AND CORRELATED MEASUREMENT NOISE TO CENTER 16X16 PIXEL DATA
233 C DO 9 J=1,16
234 C DATA(I,J)=DATA(I,J)+DATA(I+4,J+4)+DATA(I+4,J)+DATA(I,J+4)
235 C CONTINUE
236 C AND UNCORRELATED NOISE TO MEASUREMENT DATA OUTSIDE CENTER
237 C EXE PIXEL AREA
238 C CALL NOISE(DA4,512)
239 C DO 6 I=1,16
240 C DO 5 J=1,16
241 C IF(I*G+1)/LE<16.AND.J*G+1/LE<16.) GO TO 6
242 C CAE(I,J)=DATA(I,J)+DATA(I,J)+DATA(I+1,J)+DATA(I,J+1)
243 C CONTINUE
244 C CREATE THE MEASUREMENT VECTOR FOR THE FILTER UPDATE
245 C 250
246 C DO 101 I=1,16
247 C DO 101 J=1,16
248 C 251
249 C IF(I*G+1)(CASEL+DATA*I+J*Y)+CASE+1)
250 C 211(CASE+1) CASE=CASE+1
251 C 211(CASE+1) CASE+1
252 C GO CALCULATE THE ERRORS OF THE FILTER'S ESTIMATE PRIOR TO
253 C MEASUREMENT INCORPORATION
254 C CALL STATMF(XFRE,XFRE+2,CRRE+2,CRRE+2,XFRE+2,XT+Y+XR)
255 C INTEGRATION MEASUREMENT
256 C IF(IFN+1) GO TO 164
257 C CALL UPKF(XFRE+2,CRRE+2,CRRE+2,CRRE+2,XT+Y+XR)
258 C CALCULATE THE ERRORS FOR THE FILTER AFTER THE INCORPORATION
259 C OF THE MEASUREMENT
260 C CALL STATMF(XFRE+2,CRRE+2,CRRE+2,CRRE+2,XT+Y+XR)
COMPUTE THE SHIFT INFORMATION FROM THE CENTER OF FOV

YSHIFT = YFP(2) + 4 = YFP(4)

SHIFT THE DATA APPROPRIATELY

GET FORWARD

164 CALL FOUR (DATA, NN + 2, -1, I, WORK)

FILTER DESIRED FREQUENCY COMPONENTS OUT

IF (IFREQ GT 117) IFREQ = 12

DO I = 1, IF, 1

IC = 2, I, 24

DATA(I, I) = CMPLX(0, 0)

DATA(J, I) = CMPLX(0, 0)

CONTINUE

ASSURE IF L=1 THAT THE DATA IS CENTERED

IF (K = 11) CALL SHIFT (DATA, 24, XSHIFT, YSHIFT)

CALL SMOOTH DATA, SDLT, ALPHA + 24, NR

CALL PROPFF (PHI, GFD, PFP, PFM, XFP, XFM)

PROPAGATE TRUTH MODEL STATE ONE FRAME

CALL PROP (PH, GFD, PFP, PFM, XFP, XFM)

CONTINUE

END MONTE CARLO SIMULATION

CALCULATE MEAN AND VARIANCE STATISTICS

CALL FLSTXFMM, XFM2, CNME, CNME2, XFPE, XFP2, CNPE, CNPE2, NRUNS

WRITE (4, 9987) NRUNS, NFAMES, NZ, IFREQ, COV, VARP, ALPHA

9987 FORMAT (11, 10, 5Pe12.3, 2G12.3, 2F12.3, 15F12.3, 8F12.3)

*TRUTH MODEL UNCERTAINTY = 1.7355

GO TO 5421

STOP

END
(0

to
I

I

":z

LjJ 4.T

Z

It

if

U
p.:

)4

toI

j

4.,

a

V* c.

Qr

t,

3.

t

LO

w
W'. r

c

N't9

259
4.*

_TM.
I,

.

t.

NV,

x x

"

u>

I


Plotting Commands
ENHANCED TRACKING OF AIRBORNE TARGETS USING FORWARD LOOKING INF--ETC(U)

DEC 81  S K ROGERS

AFIT/GE0/EE/81D-5

END DATE 7-82

7-82
Appendix I

Plots of Tracking Errors

The sequence of these plots is explained in Chapter VI. This appendix contains plots for four settings of parameters which control intensity function derivation and Kalman filter characteristics. The legend on each plot provides information on the target's spread parameter, the number of zeros which are padded, the smoothing constant, the number of frequencies to zero, the background noise strength, the standard deviation of the discrete time noise driving the target dynamic states of the filter respectively.
Figure 1-2. Pad Zeros Y Minus Errors
Figure 1-3. Pad Zeros X Plus Errors
Figure I-4. Ped zeta vs Y plus error.

Estimated Y Plus Position Mean Error +/- Sigma

Error (Pixels)

Frame

Copy No. AF-IF-080.293 = 0.5.1.01..11
Figure 1-5. Pad Zeros X Position Errors
Figure 1-6. Pad Zeros Y Position Errors
Figure 1-9. Pad Zeros X Centroid Plus Errors
Figure 1-10. Pad zeros y centroid plus errors.
Figure 1-11. Pad Zeros X Centroid Position Errors
Figure 1-12. Pad Zeros Y Centroid Position Errors
Figure 1-13. Pad Errors of Estimated h
Figure 1-14. Pad Zeros Error of Estimated Dh/Dx
Removing Two Highest Spatial Frequencies
Figure 1-16. Removing Two Highest Spatial Frequencies
X Minus Errors
Figure 1-18. Removing Two Highest Spatial Frequencies
X Plus Errors
Figure 1-19. Removing Two Highest Spatial Frequencies Y Plus Errors
Figure 1-20. Removing Two Highest Spatial Frequencies
X Position Errors
Figure 1-21. Removing Two Highest Spatial Frequencies, Y Position Errors.
Figure 1-22. Removing Two Highest Spatial Frequencies X Centroid Minus Errors
Figure 1-23. Removing Two Highest Spatial Frequencies Y Centroid Minus Errors
Figure 1-24. Removing Two Highest Spatial Frequencies
X Centroid Plus Errors
Figure 1-25. Removing Two Highest Spatial Frequencies
Y-Centroid Plus Errors
Figure 1-26. Removing Two Highest Spatial Frequencies
X Centroid Position Errors

297
Figure 1-27. Resolving Two Highest Spatial Frequencies Y Centroid Position Errors
Figure 1-28: Error of Estimated Intensity Profiles.
Figure 1-29. Removing Two Highest Spatial Frequencies
Error of Estimated DH/Dx
Figure 1-30. Removing Two Highest Spatial Frequencies
Error of Estimated Dh/Dy
Removing Four Highest Spatial Frequencies
Figure 1.31. Removing Four Highest Spatial Frequencies
X Minus Errors
Figure 1-32. Removing Four Highest Spatial Frequencies Y Minus Errors
Figure 1.4:
Removing four principal spatial frequencies

ESTIMATED X PLUS POSITION MEAN ERROR +/- SIGMA

ERROR (PIXELS)

FRAME

COV. WZ, A, R, AF; VAR. SE3 = 0.0, 1.4, 1.1
Figure 1-34. Removing Four Highest Spatial Frequencies Y Plus Errors
Figure 1.35. Removing Four Highest Spatial Frequencies
X Position Errors
Figure 4.9b. Removing Four Highest Spatial Frequencies
Y Position Errors
Figure 147. Removing Four Highest Spatial Frequencies X Centroid Minus Error.
Figure 1-38. Removing Four Highest Spatial Frequencies
Y Centroid Minus Errors
Figure 1-40. Removing Four Highest Spatial Frequencies
Y Centroid Plus Errors
Figure 1-11. Removing Four Highest Spatial Frequencies 
X Centroid Position Errors
Figure 1-42. Removing Four Highest Spatial Frequencies Y Centroid Position Errors
Figure 1.43. Removing Four Highest Spatial Frequencies Error of Estimated h
Figure 1-44. Removing Four Highest Spatial Frequencies Error of Estimated Dh/Dx
Maximum Noise
ESTIMATED X MINUS POSITION MEAN ERROR +/- SIGMA
Figure 1-47. Maximum Noise Y Minus Errors
Figure 1.48. Maximum Noise X Plus Errors
Figure 4.49. Maximum Noise Y plus Errors
Figure 1-50. Maximum Noise X Position Error
Figure 1-51. Maximum Noise Y Position Errors
Figure 1.52. Maximum Noise X Centroid Minus Errors
Figure 1-51. Maximum Noise Y Centroid Minus Errors
Figure 1-54. Maximum Noise X Centroid Plus Errors
Figure 1-55. Maximum Noise Y Centroid Plus Errors
Figure 1-56. Maximum Noise X Centroid Position Errors
Figure 1-57. Maximum Noise Y Centroid Position Errors
Figure 1-58. Maximum Noise Error of Estimated h
Figure L-59. Maximum Noise Error of Estimated $Dh/Dx$
Figure 1.69: Mean Error +/- Sigma of Estimated DH/DY

ERROR (INTENSITY)

FRAME

1.00 0.50 0.00 -0.50 -1.00 -1.50 -2.00 -2.50 -3.00 -3.150

1.00 0.50 0.00 -0.50 -1.00 -1.50 -2.00 -2.50 -3.00 -3.150

1.00 0.50 0.00 -0.50 -1.00 -1.50 -2.00 -2.50 -3.00 -3.150
Appendix J
Extended Kalman Filters

This appendix provides an overview of the generic extended Kalman filter algorithm (17). The generic terms of the algorithm are then presented in terms of the tracking example.

The system which contains the states which are to be estimated is assumed to satisfy the nonlinear stochastic differential equation

\[ \dot{x}(t) = f[x(t), u(t), t] + G(t) \, w(t) \]  \hspace{1cm} (J-1)

where the initial condition on the state, \( x(t_0) \), is modelled as a Gaussian random vector with mean \( \hat{x}_0 \) and covariance \( P_0 \).

The dynamic driving noise \( w(t) \) is a zero-mean white Gaussian noise process of strength \( Q(t) \).

\[ \mathbb{E}\{w(t) \, w^T(t+\tau)\} = Q(t) \, \delta(\tau) \]  \hspace{1cm} (J-2)

The dynamic driving noise, \( w(t) \), is assumed to enter in a linear additive fashion in Equation (J-1).

For this application, the generic Equation (J-1) becomes Equation (4-4) where the generic \( f \) is replaced by a constant \( F \). The four filter states, target position in x-y coordinates resulting from dynamics or atmospherics \( [x_d \, y_d \, x_a \, y_a]^T \), are multiplied by the constant filter plant matrix \( F_f \).

The generic discrete-time measurement equation is
modelled as a nonlinear function of the states and time plus linearly additive noise corruption

\[ \mathbf{z}(t_i) = h[\hat{\mathbf{x}}(t_i), t_i] + \mathbf{v}(t_i) \]  

(J-3)

The measurement corruption noise \( \mathbf{v}(t_i) \) is a white zero-mean Gaussian noise of strength \( \mathbf{R}(t_i) \) and independent of the dynamic driving noise \( \mathbf{w}(t_i) \) and the initial conditions on the states.

\[
E(\mathbf{v}(t_i) \mathbf{v}^T(t_i)) = \begin{cases} \mathbf{R}(t_i) & t_i=t_j \\ 0 & t_i \neq t_j \end{cases}
\]  

(J-4)

In this application the measurement vector, \( \mathbf{z}(t_i) \) is composed of the 64 intensity measurements from the FLIR. The \( h[\cdot,\cdot] \) function is the expected intensity measurements as a function of the estimated states. The derivation of this intensity function is the subject of Chapter II.

The propagation and measurement update equations for the extended Kalman filter are presented in Chapter IV. These equations are coupled to the state estimate relations via the dependence of \( h[\cdot,\cdot] \) on the state estimate \( \hat{\mathbf{x}}(t_2) \).
VITA

Steven Keith Rogers was born April 6, 1953 in Jackson, Mississippi. He graduated from E.E. Smith High School in June 1971. In May 1978, he graduated from the University of Colorado with Bachelor degrees in Electrical Engineering and also in Computer Science and entered Air Force Officers Training School that same month. From August 1978 to June 1980, he served as an electrical engineer with the Aeronautical Systems Division at Wright-Patterson Air Force Base, Ohio. In June 1980, Second Lieutenant Steve Rogers was assigned to the Air Force Institute of Technology to pursue a Master's of Science Degree in Electrical Engineering (Electro-Optics). Lt. Rogers is a member of Tau Beta Pi.
ENHANCED TRACKING OF AIRBORNE TARGETS USING FLIR MEASUREMENTS

Steven K. Rogers
Lt, USAF

Air Force Institute of Technology (AFIT/EN)
Wright-Patterson AFB, Ohio 45433

Air Force Weapons Laboratory/ARAA
Kirtland AFB NM 87117

Approved for public release; distribution unlimited

Considerable work has been accomplished at AFIT in the last three years to improve the tracking capability of the high energy laser weapon. The improvements were achieved via use of an adaptive extended Kalman filter algorithm. In this research, work is initiated on a tracker able to handle "multiple hot spot" targets, in which digital signal processing is employed on the FLIR data to identify the underlying target shape. This identified shape is...
20. ABSTRACT

then used in the measurement model portion of the filter as it estimates target offset from the center of the field-of-view. Two tracking algorithms are developed. The first algorithm uses an extended Kalman filter to process the intensity measurements from a FLIR to produce target position estimates. The second algorithm uses a linear Kalman filter to process the position estimates of an improved correlation algorithm. This algorithm is improved over standard correlators by using thresholding to eliminate poor correlation information, dynamic information from the Kalman filter and it also uses the on-line derived target shape.