Durham University

Durham E-Theses

The deep seismic structure of northern England and adjacent marine areas from the caledonian suture seismic project

Lewis, Anthony H. J.

How to cite:

Lewis, Anthony H. J. (1986) *The deep seismic structure of northern England and adjacent marine areas from the caledonian suture seismic project*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/7088/

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE
e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107
http://etheses.dur.ac.uk

The Deep Seismic Structure of Northern England and Adjacent Marine Areas from the Caledonian Suture Seismic Project

Anthony H. J. Lewis

Volume 2

A Thesis submitted for the degree of Doctor of Philosophy at the University of Durham

The copyright of this thesis rests with the author. No quotation from it should be published without his prior written consent and information derived from it should be acknowledged.

Graduate Society
University of Durham

Dept. of Geological Sciences

February 1986

17 JUL 1986

Thesis
1986/LEW

APPENDICES - VOLUME 2

CONTENTS

Appendix A : COMMON STATION SECTION PLOTS WITH TRAVEL-TIME PICKS . . . . . .

Appendix B : COMMON SHOT SECTION PLOTS WITH TRAVEL-TIME PICKS . . . . . .

Appendix C : COMMON DISTANCE SECTIONS AND MISCELLANEOUS PLOTS . . . . . .

Appendix D : PROGRAMME LISTINGS . . . . . .

Appendix E : CSSP NATURE PAPER - BOTT ET AL 1985 . . . . .

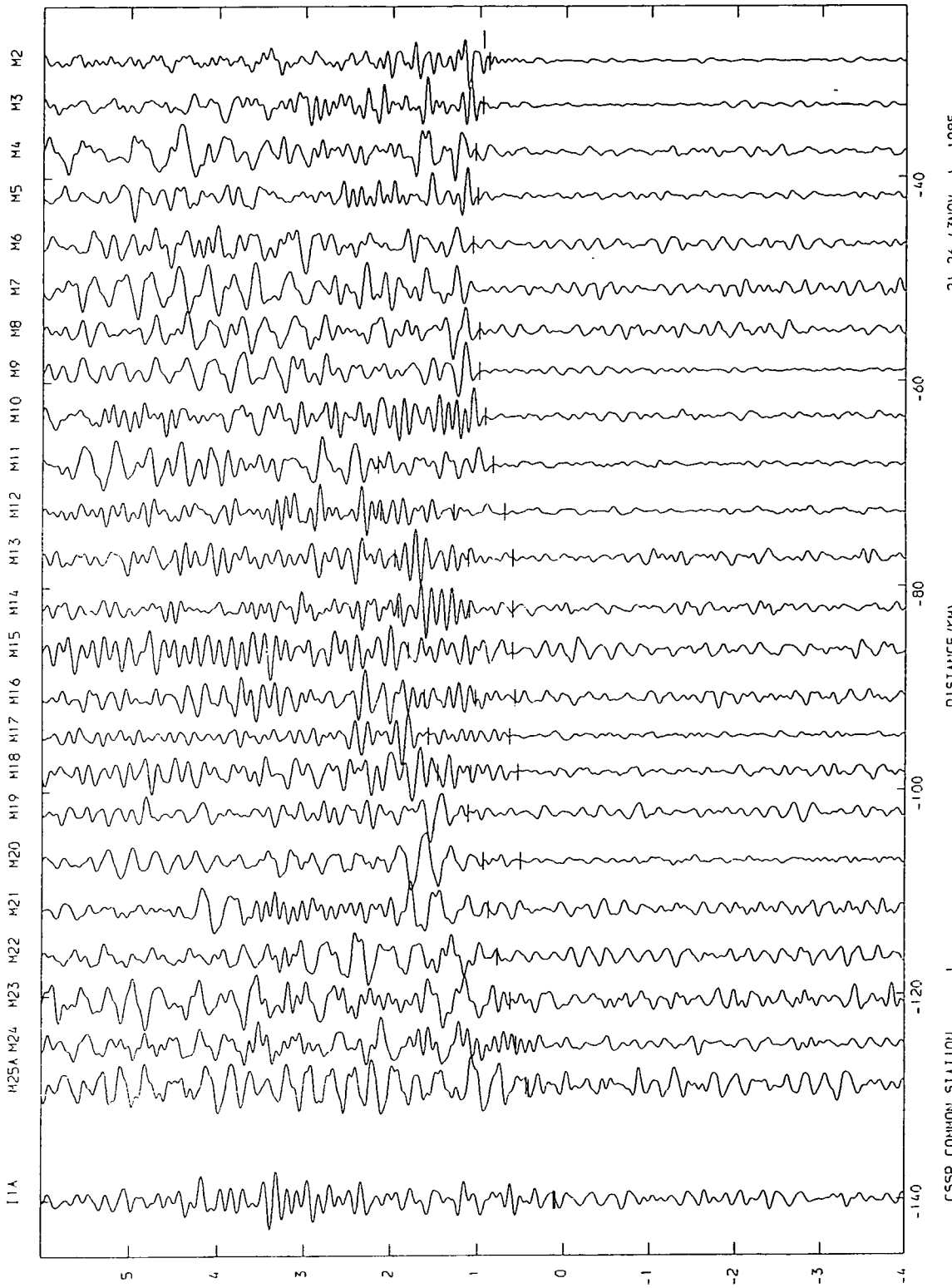
Appendix A

COMMON STATION SECTION PLOTS WITH TRAVEL-TIME PICKS

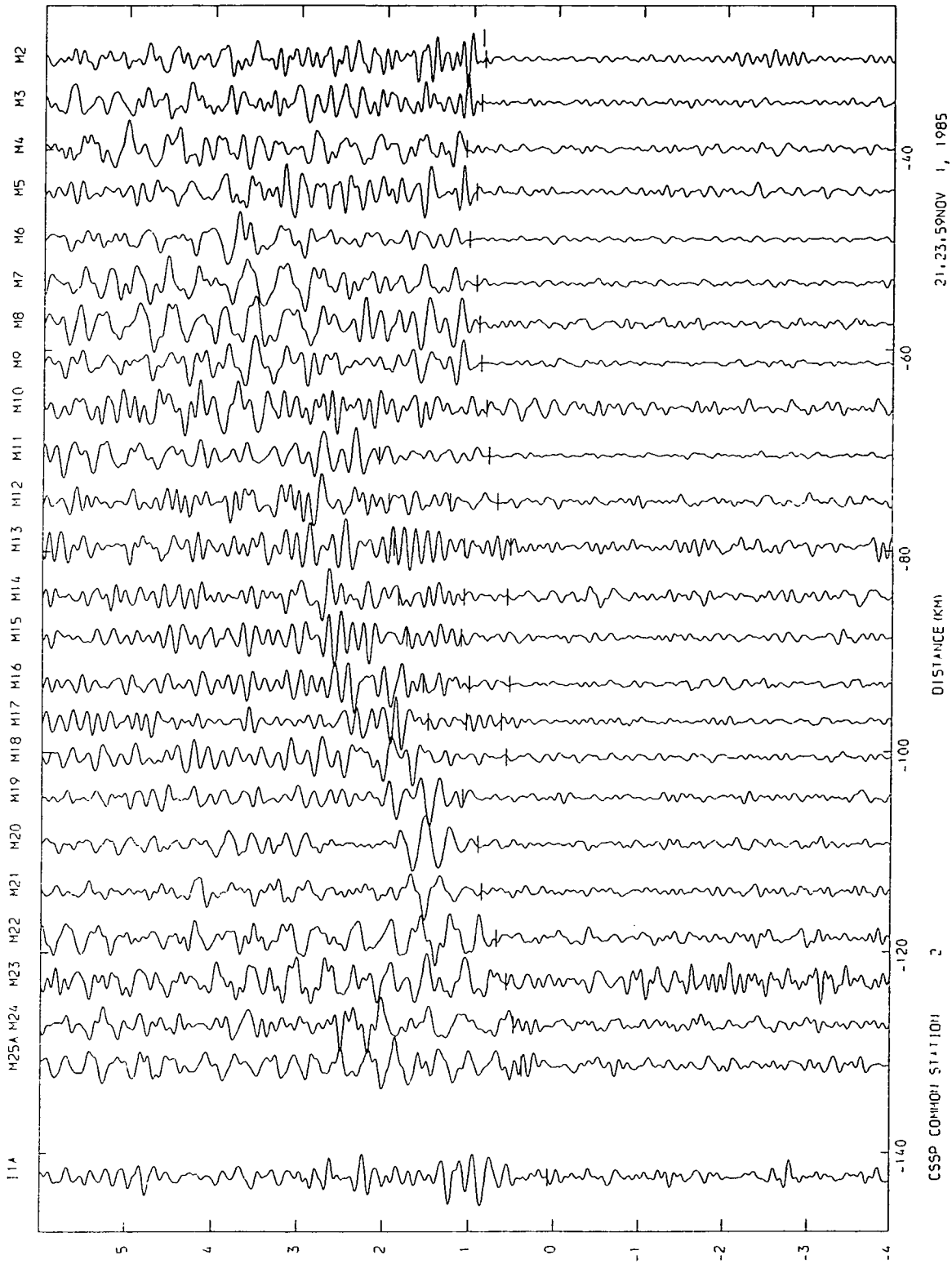
The plots provided in this appendix are all labelled with the plotting parametres and the name of the section plotted. In summary they are:

1. Common station sections for the Irish Sea shots with equalised amplitudes
2. Common station sections for the North Sea shots with equalised amplitudes
3. Common station sections for the Irish Sea shots with amplitudes corrected to common gain
4. Common station sections for the North Sea shots with amplitudes corrected to common gain

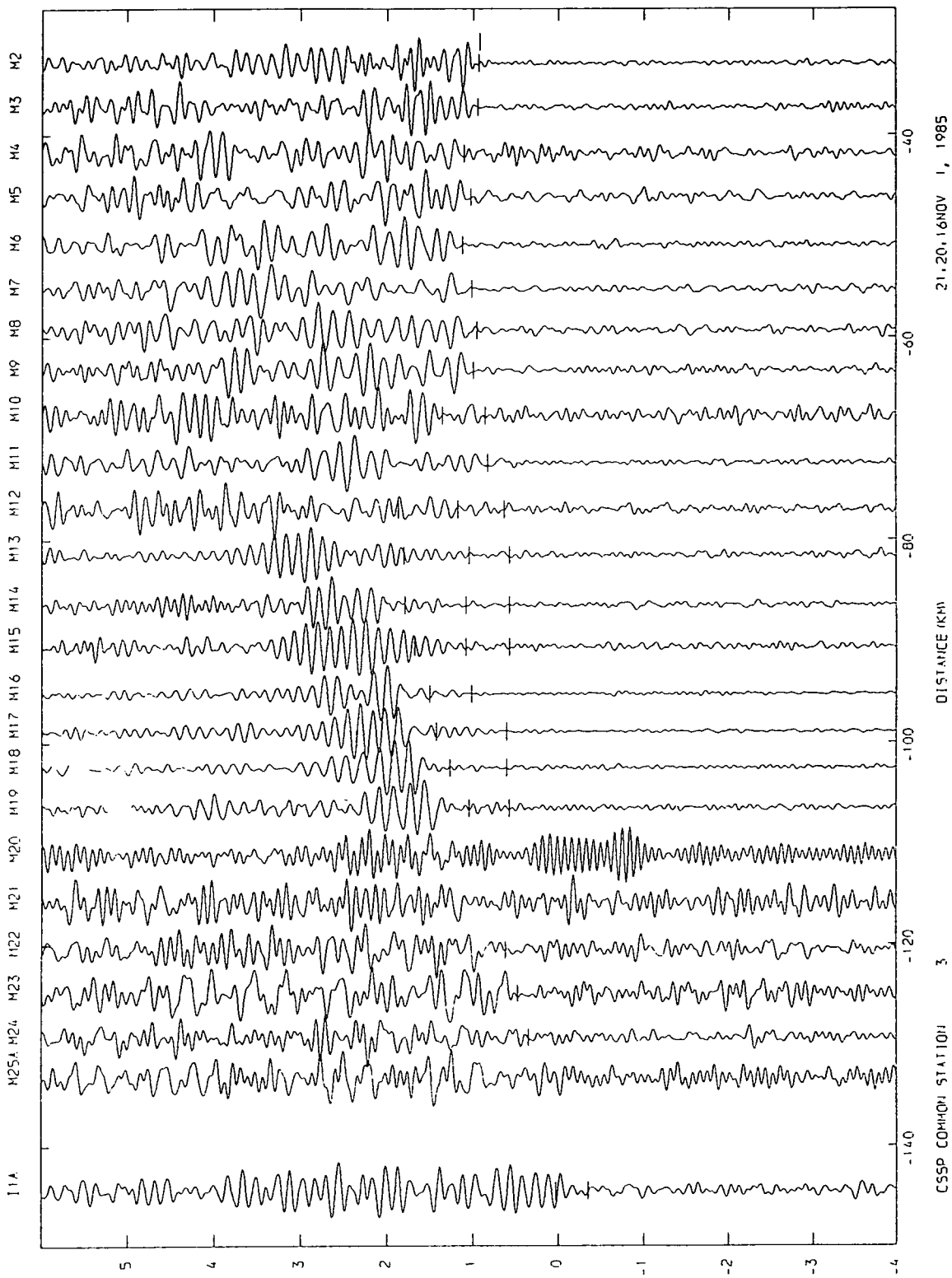
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



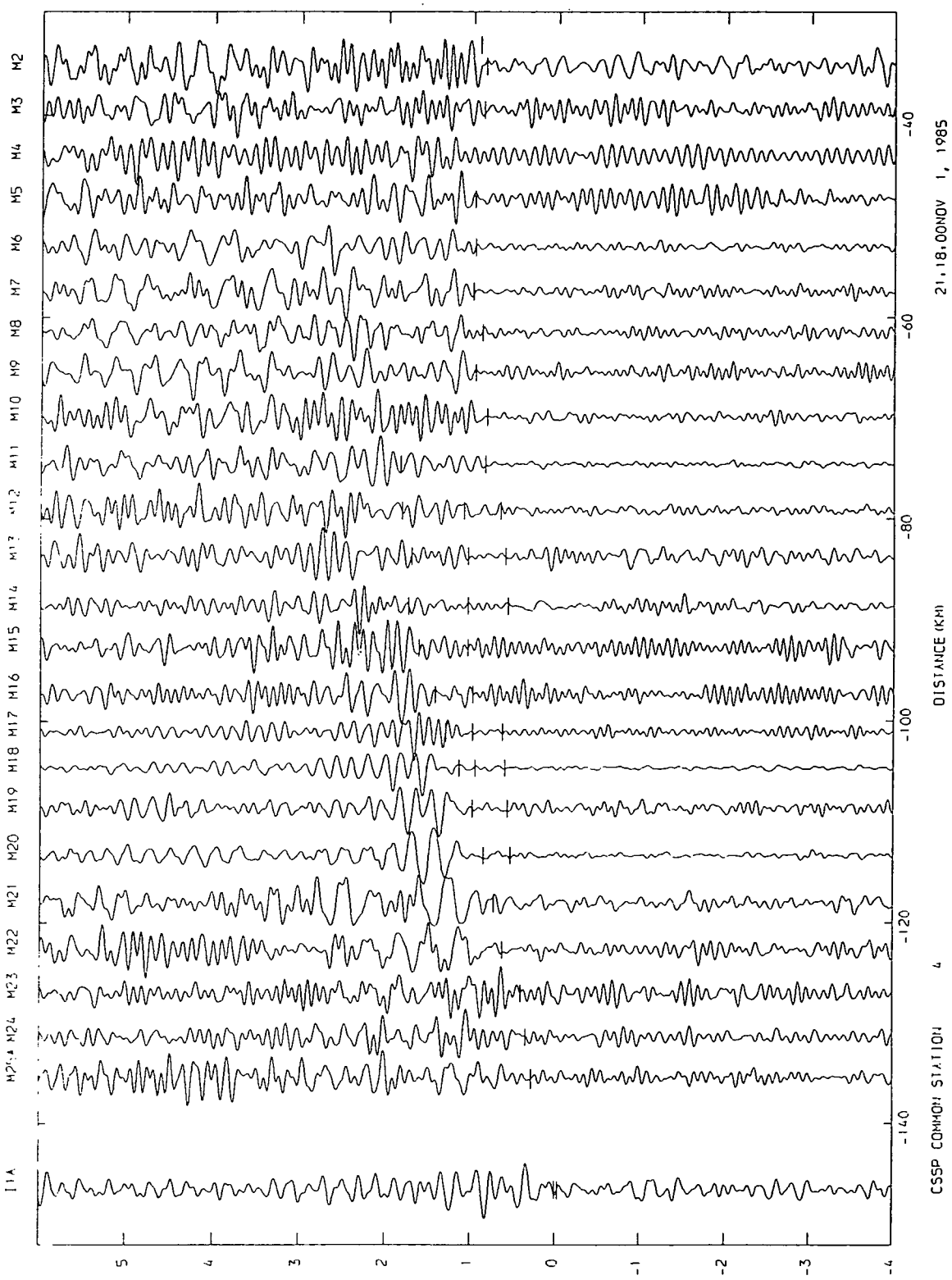
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



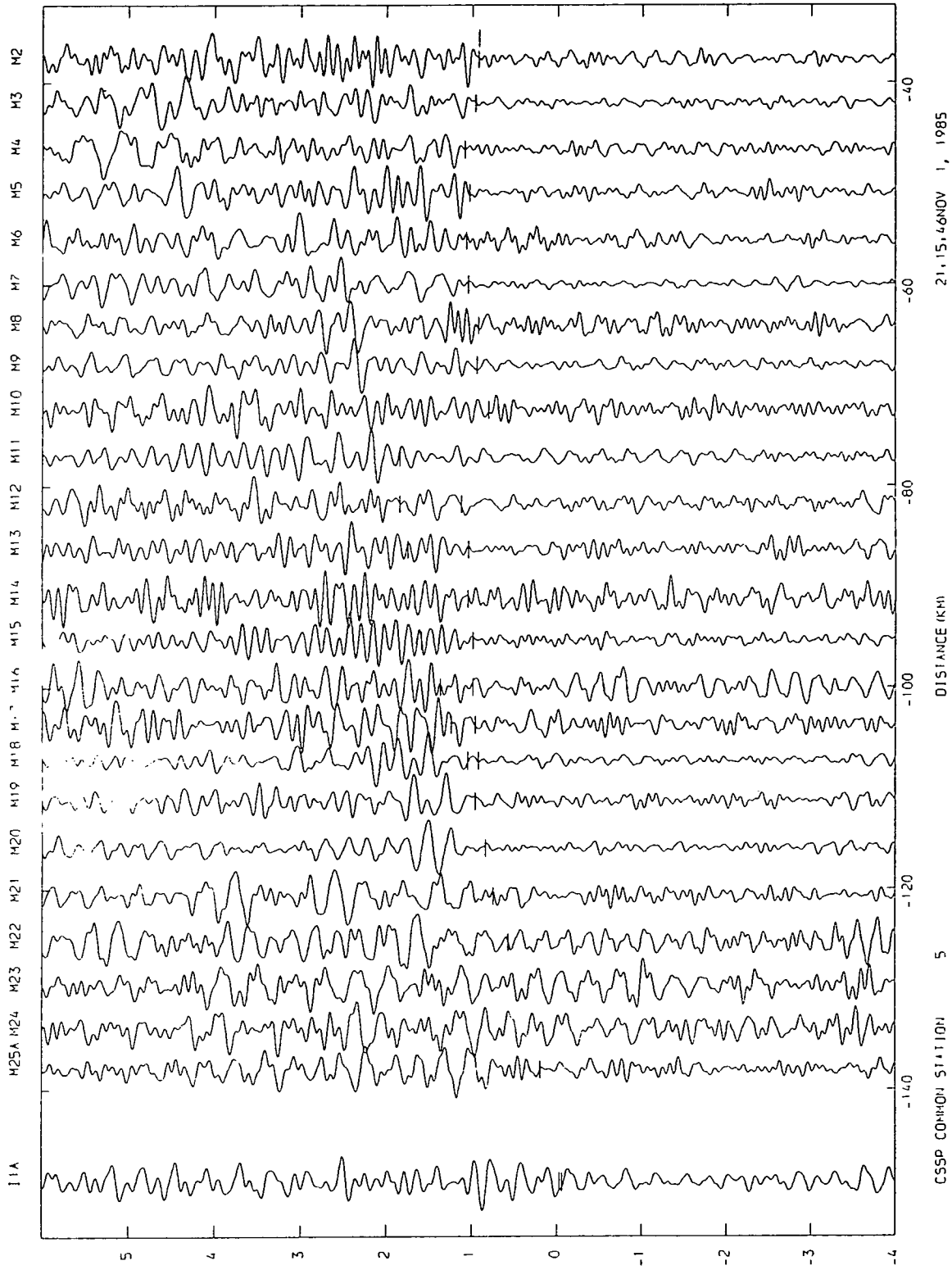
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



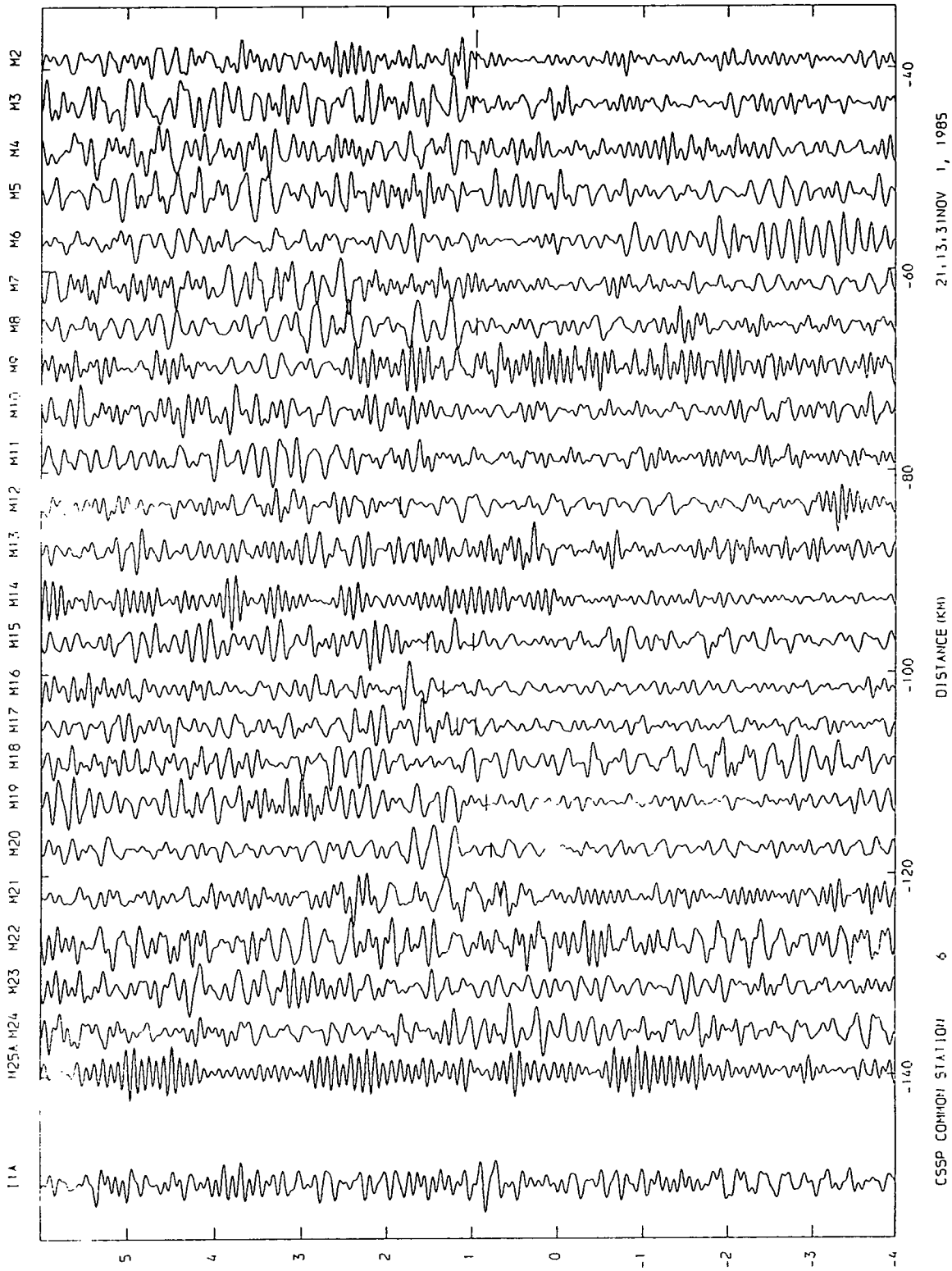
21:18:00NOV 1, 1985

CSSP COMMON STATION 4

FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



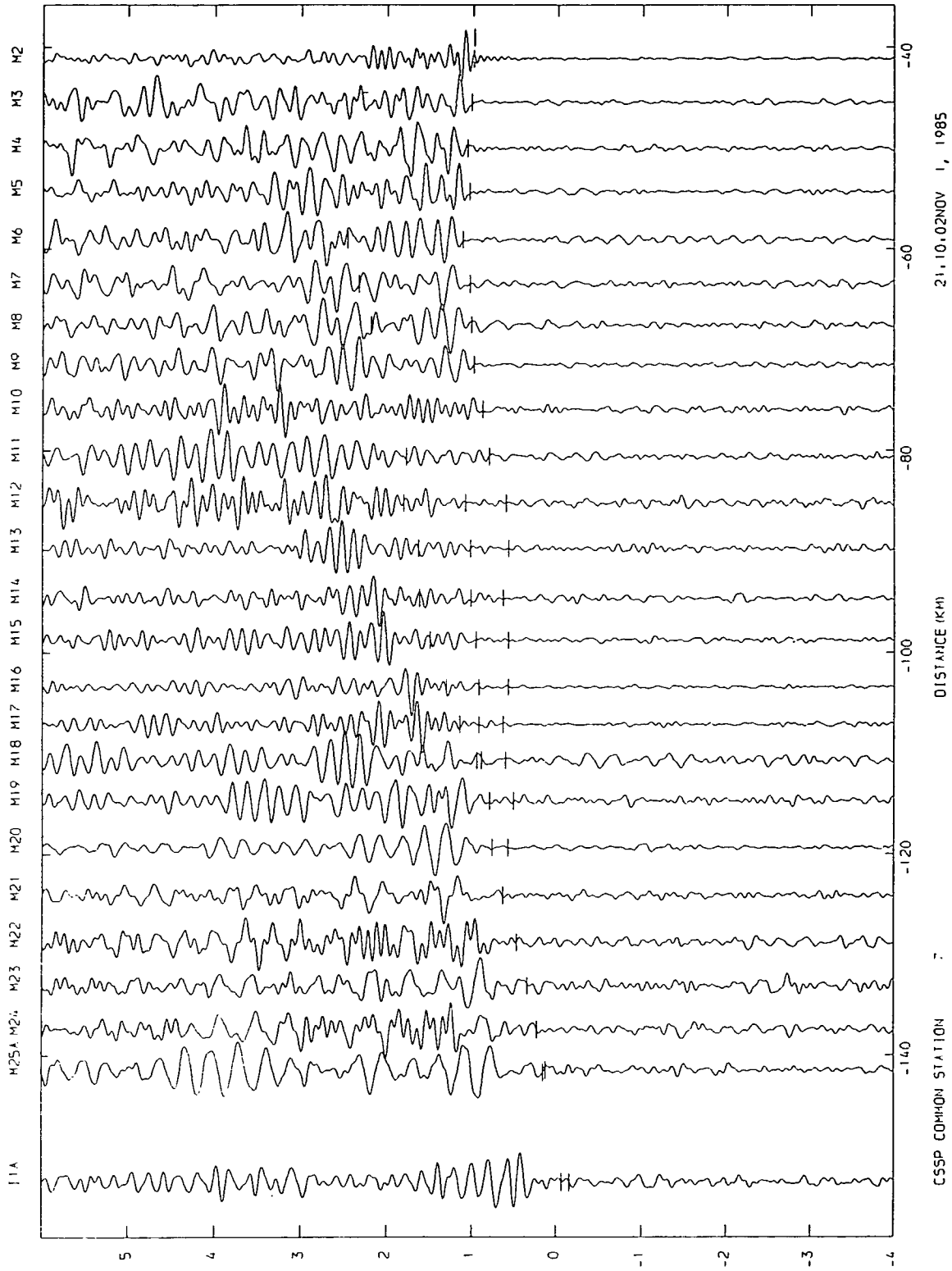
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



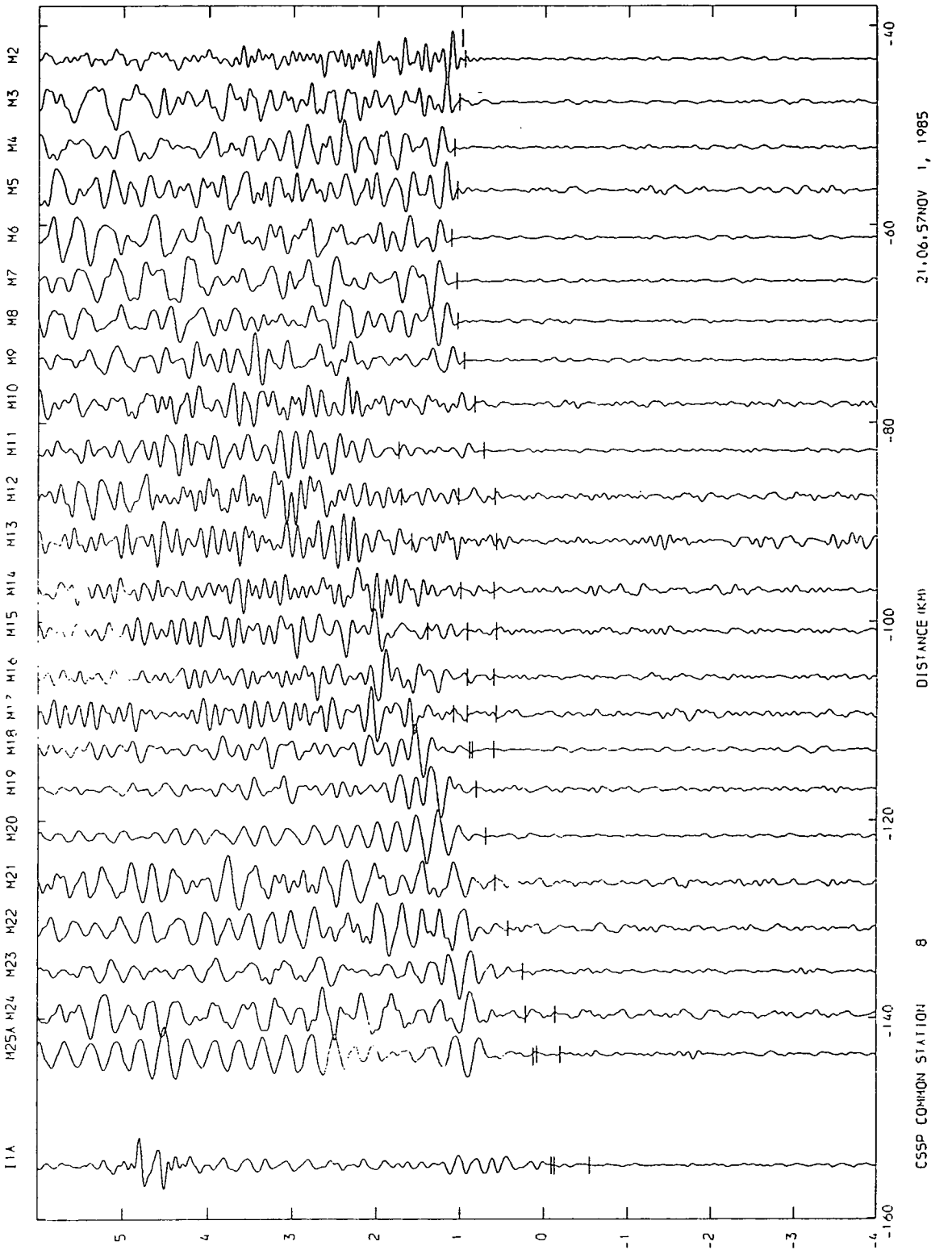
CSSP COMMON STATION 6

21,13,31NOV 1, 1985

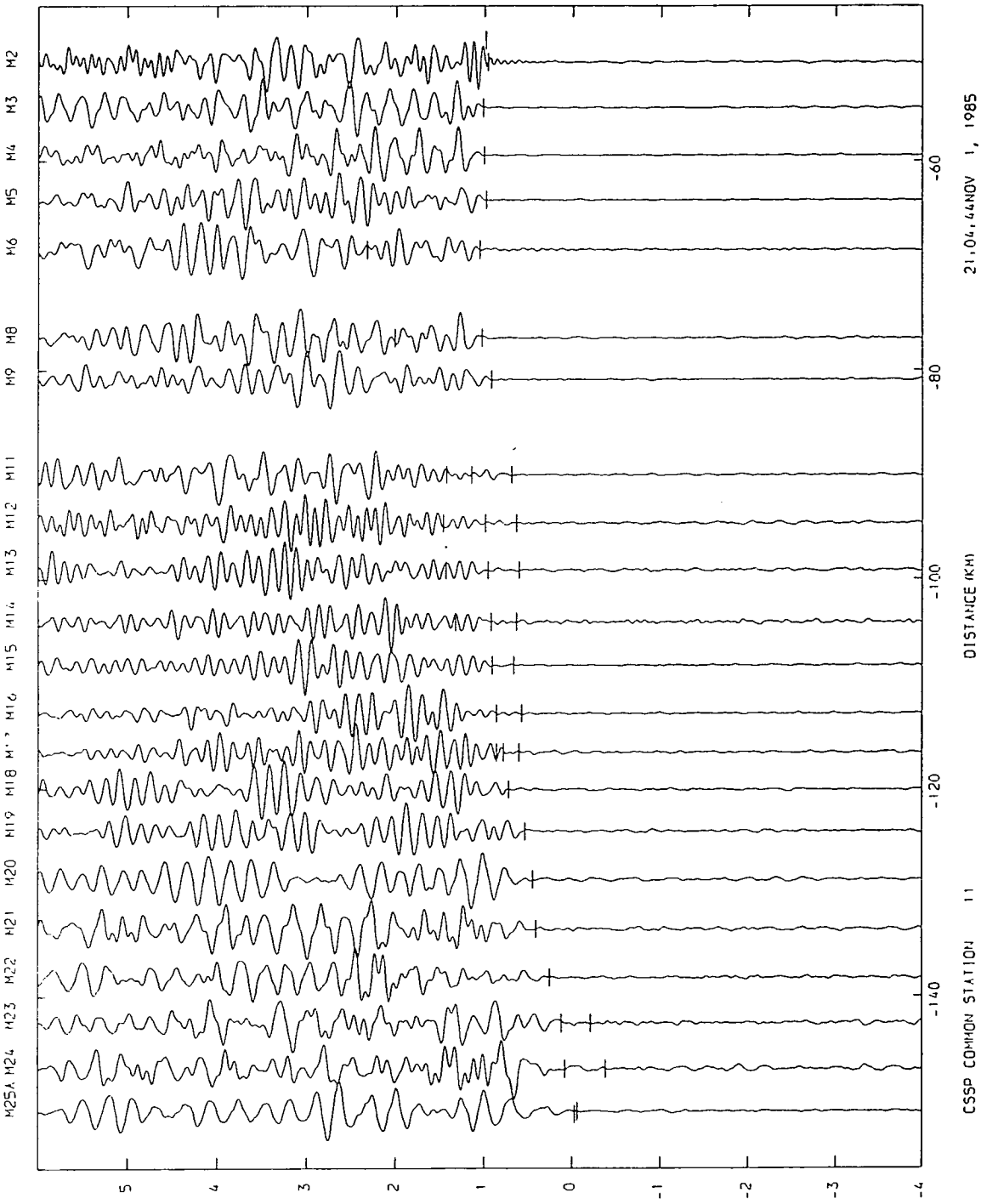
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



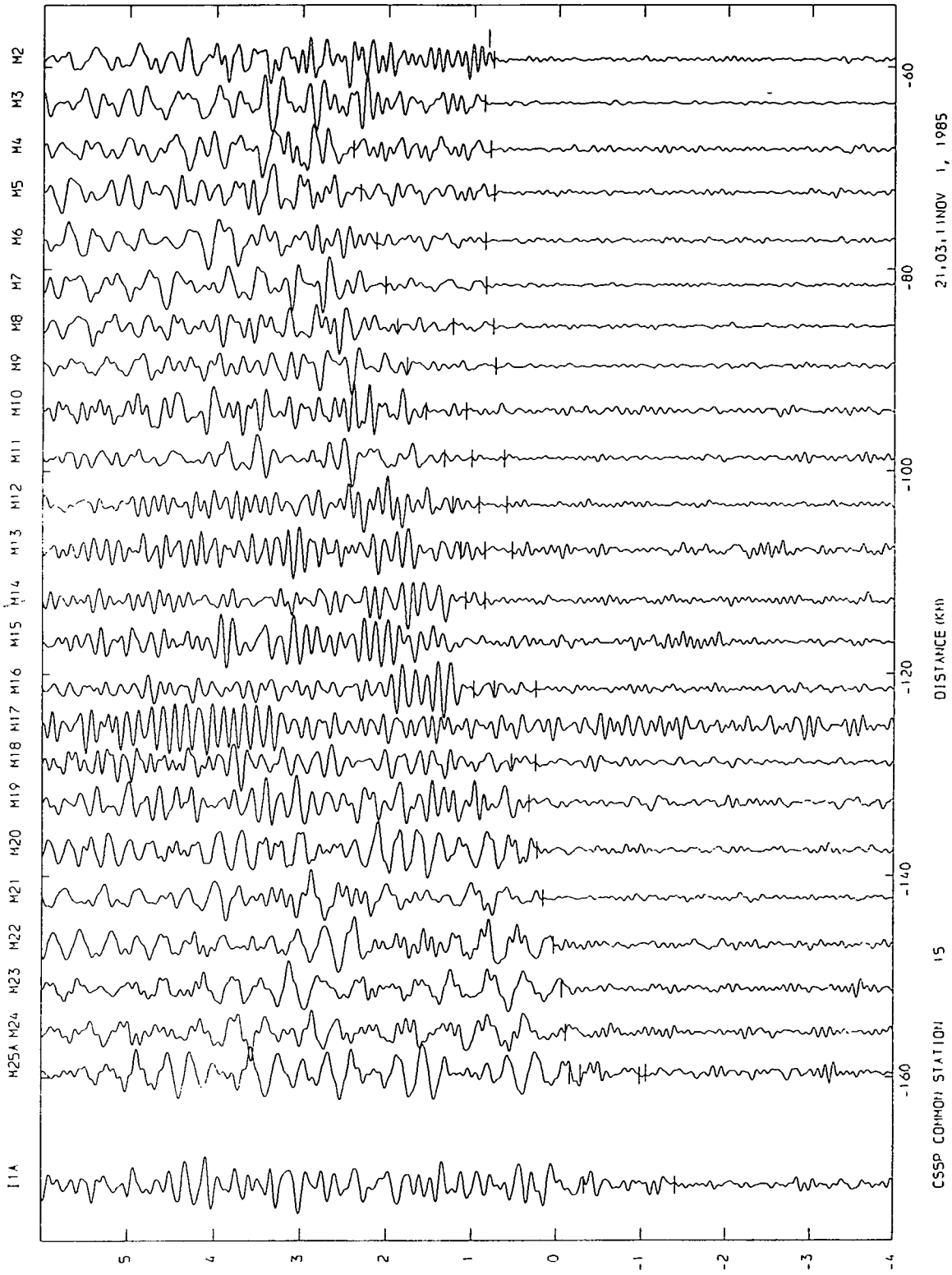
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



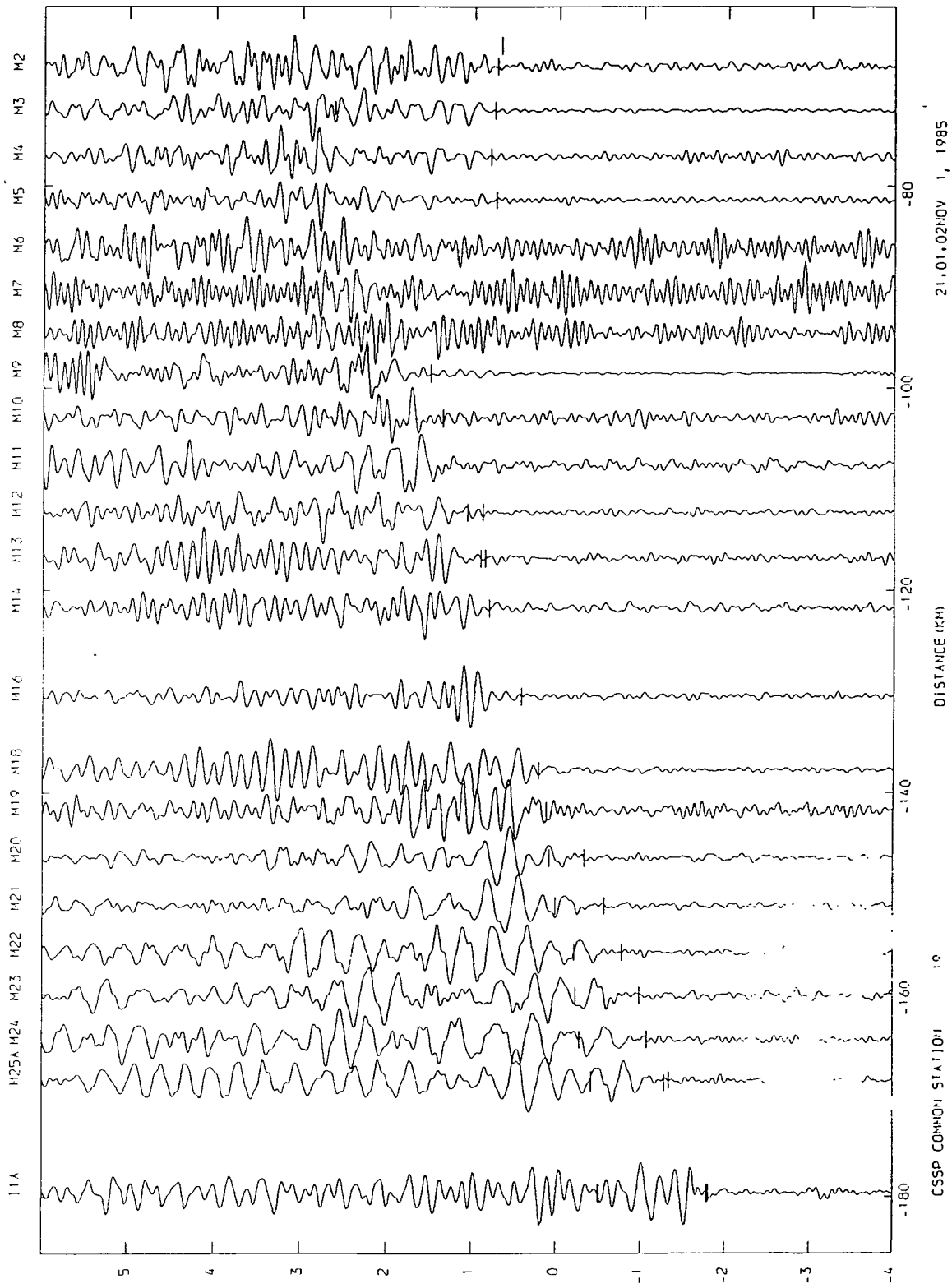
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



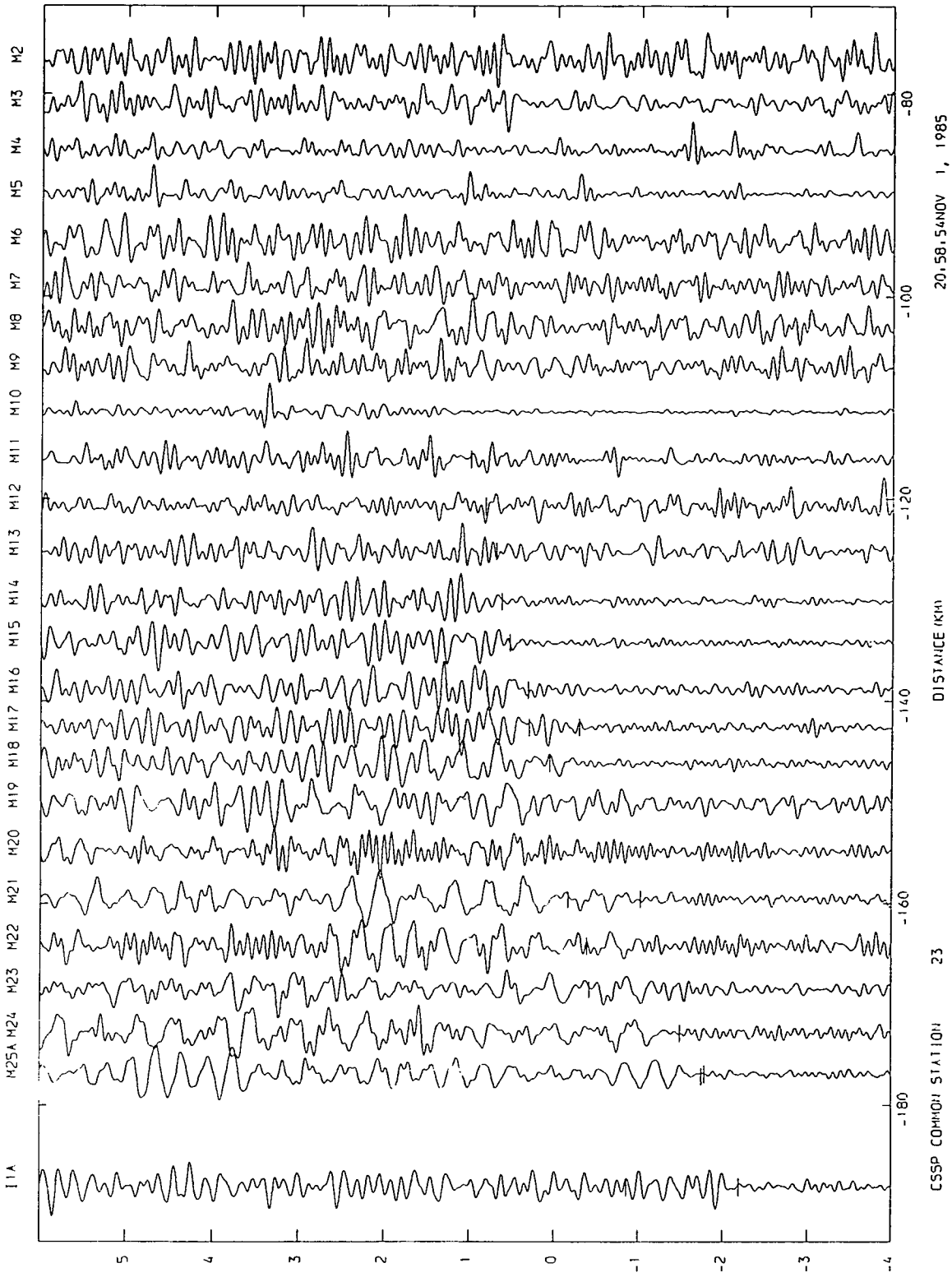
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



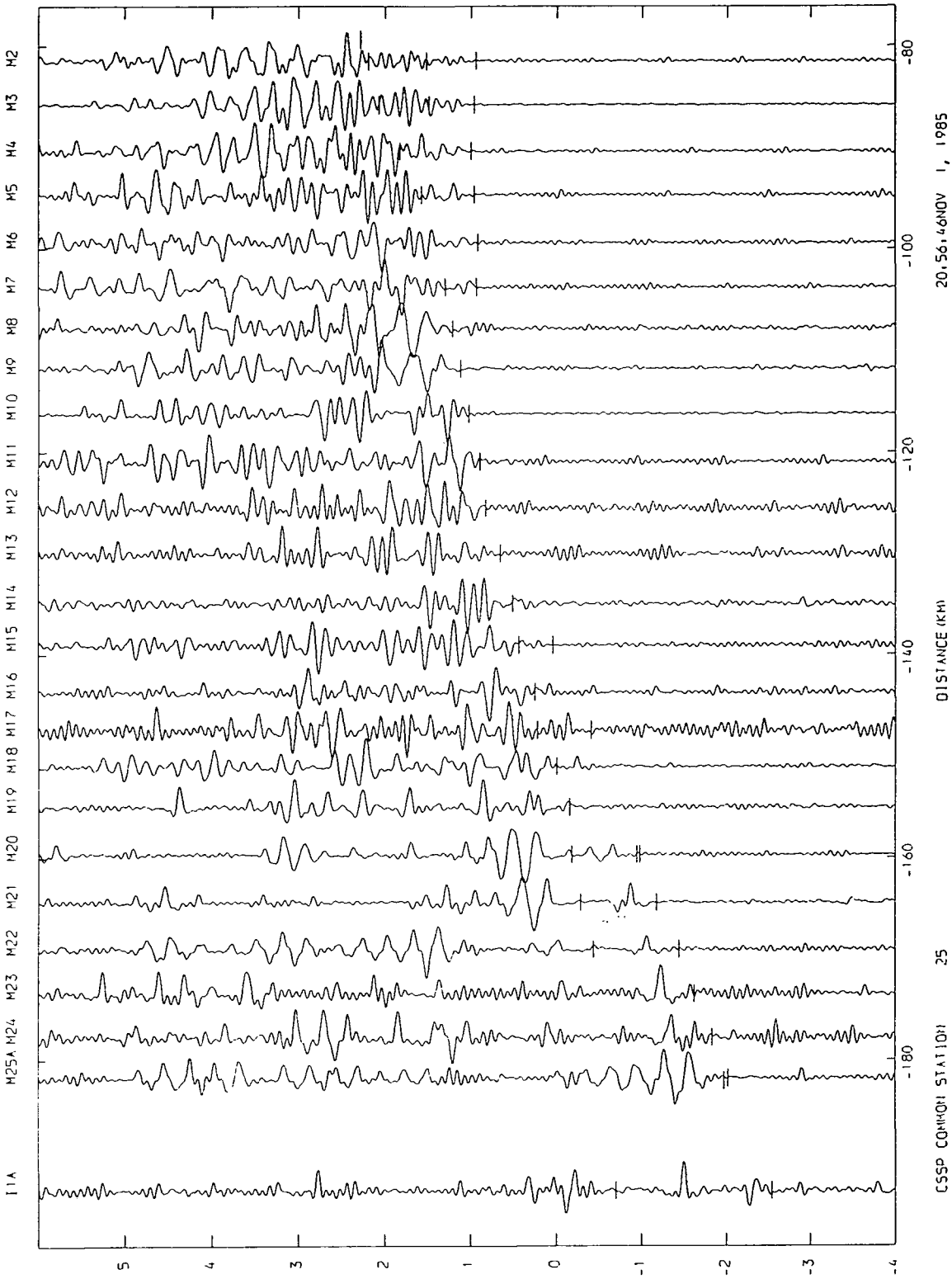
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



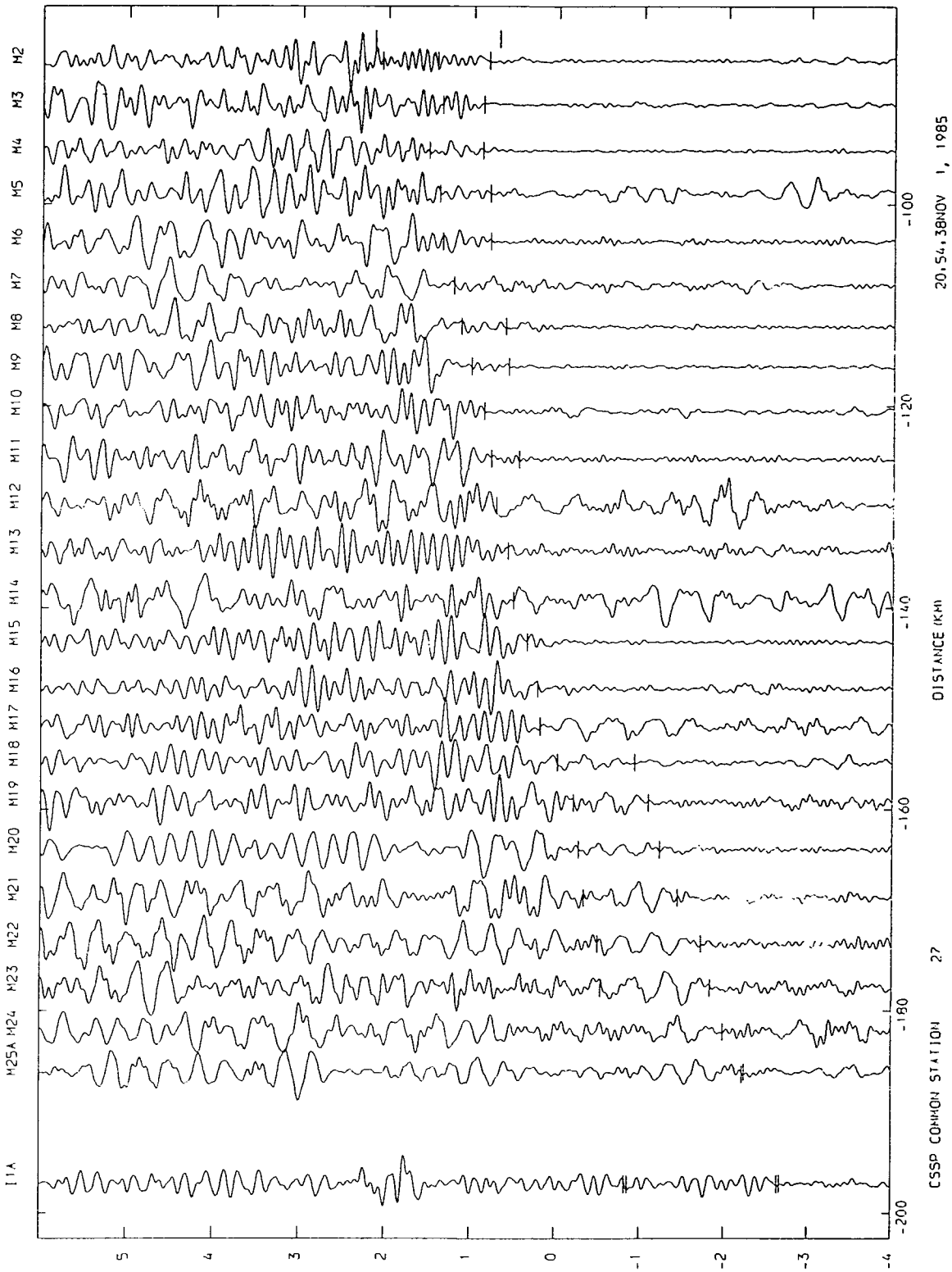
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



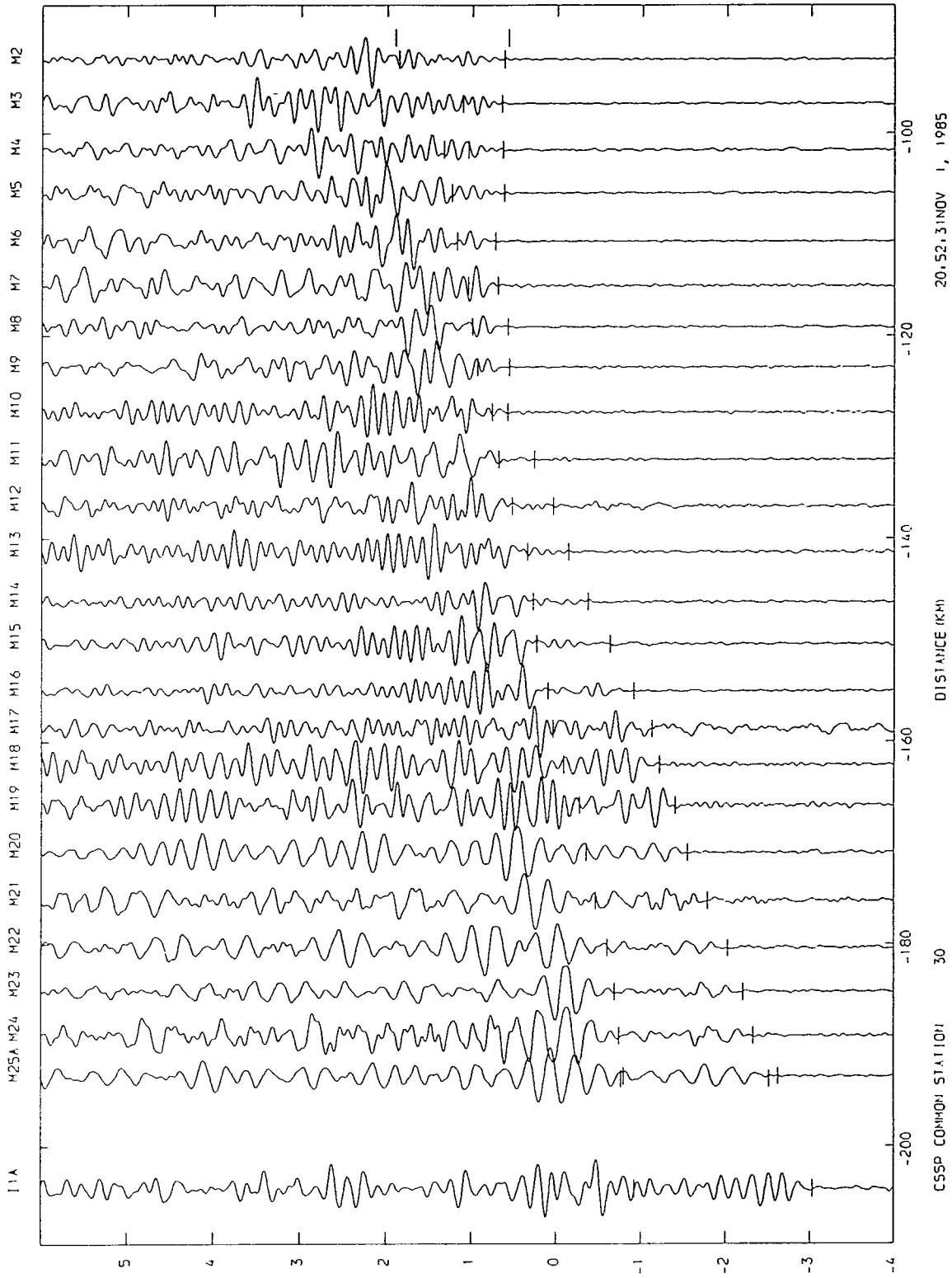
20:56:46NOV 1, 1985

CSSP COMMON STATION 25

FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

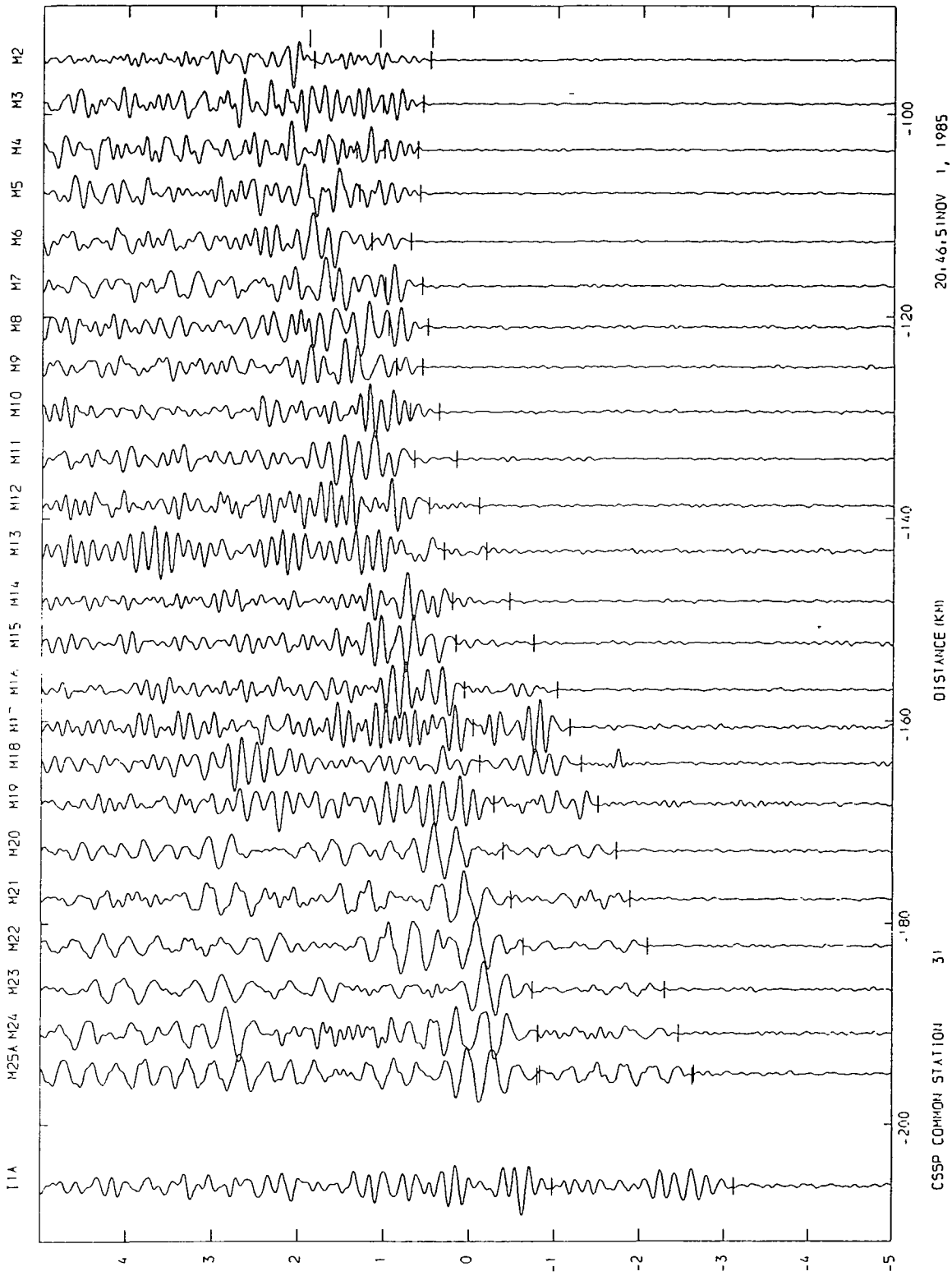


FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED

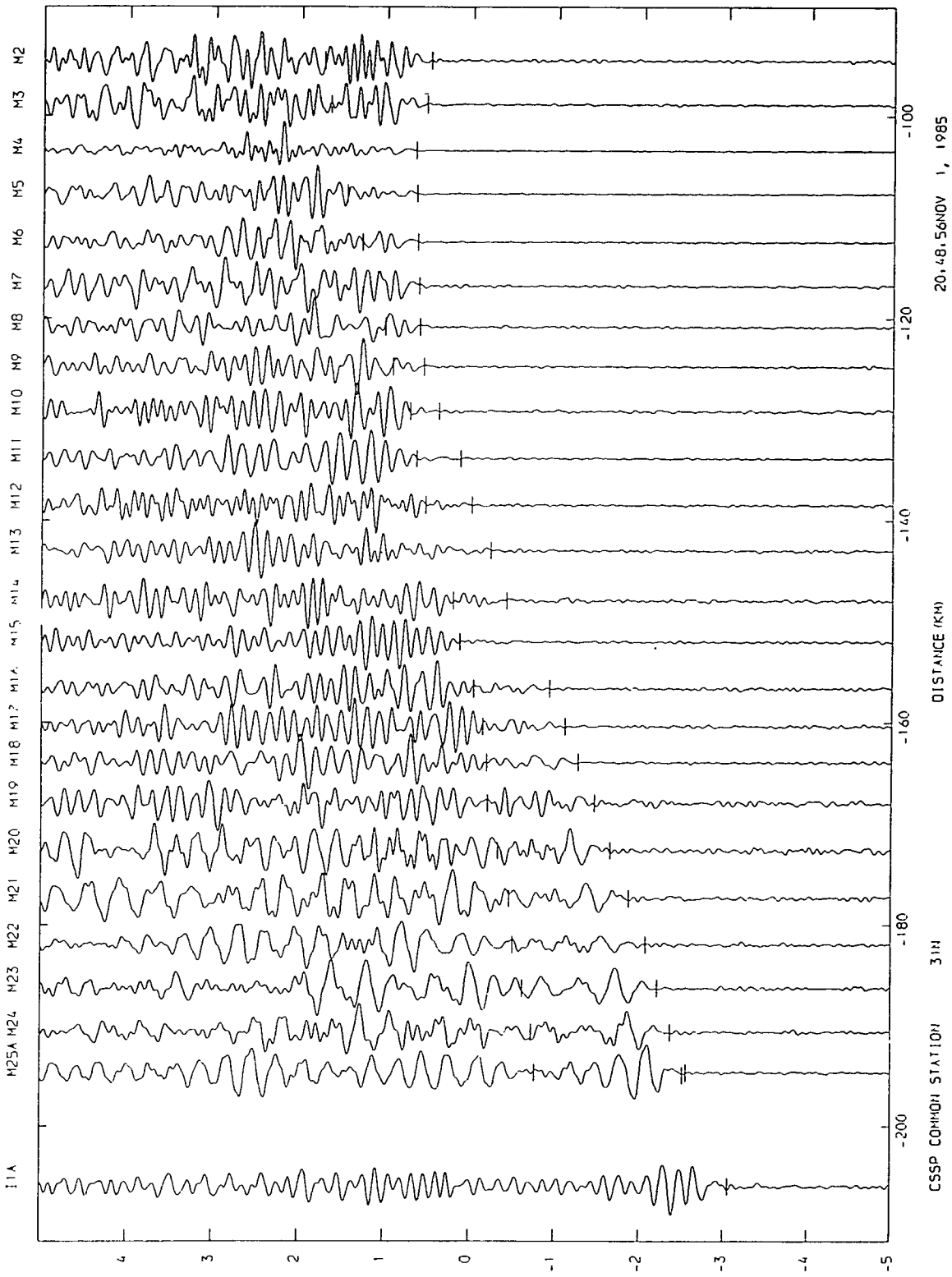


CSSP COMMON STATION 30 20:52:31NOV 1, 1985

FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



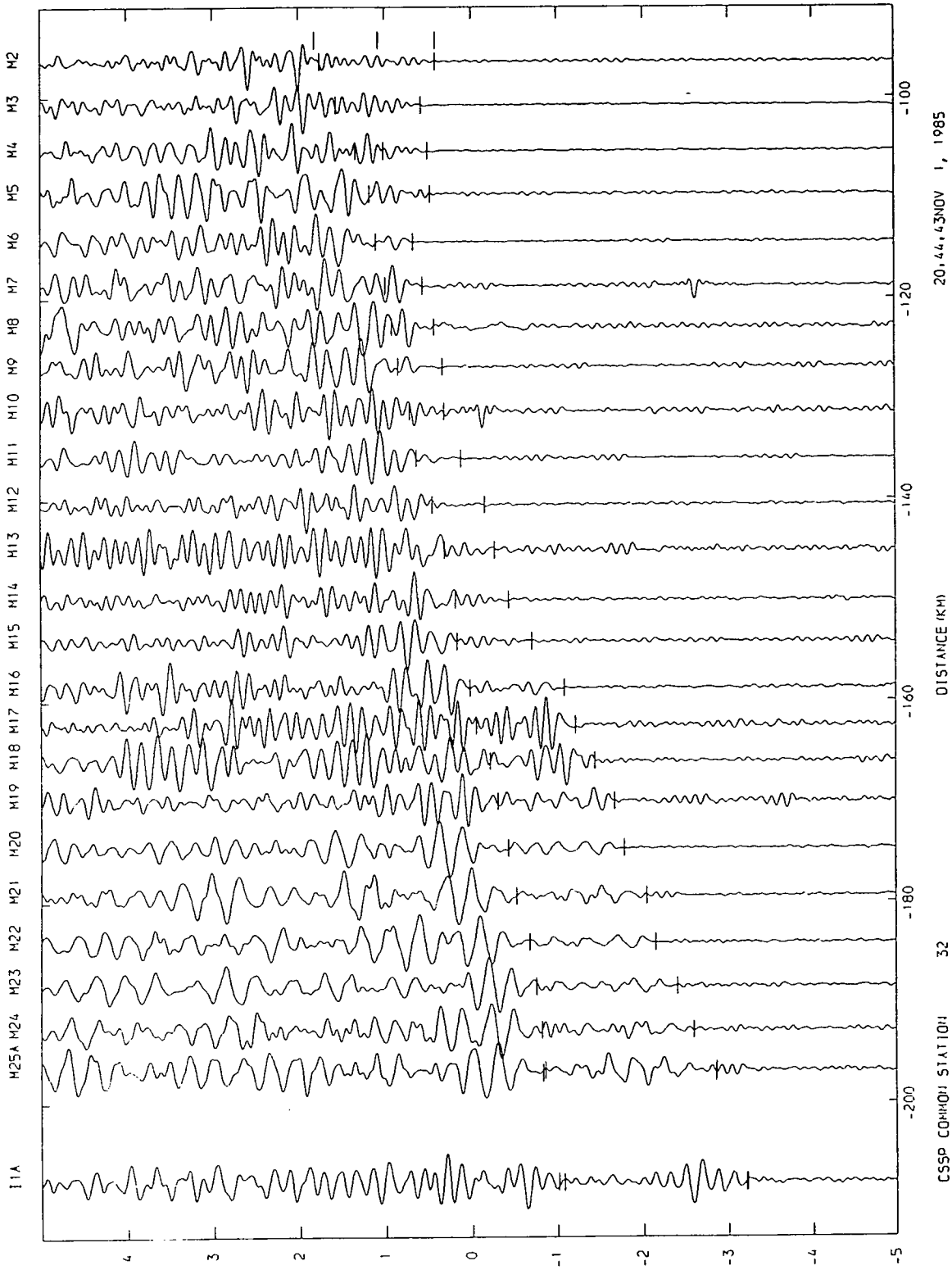
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



20:48:56NOV 1, 1985

CSSP COMMON STATION 3IN

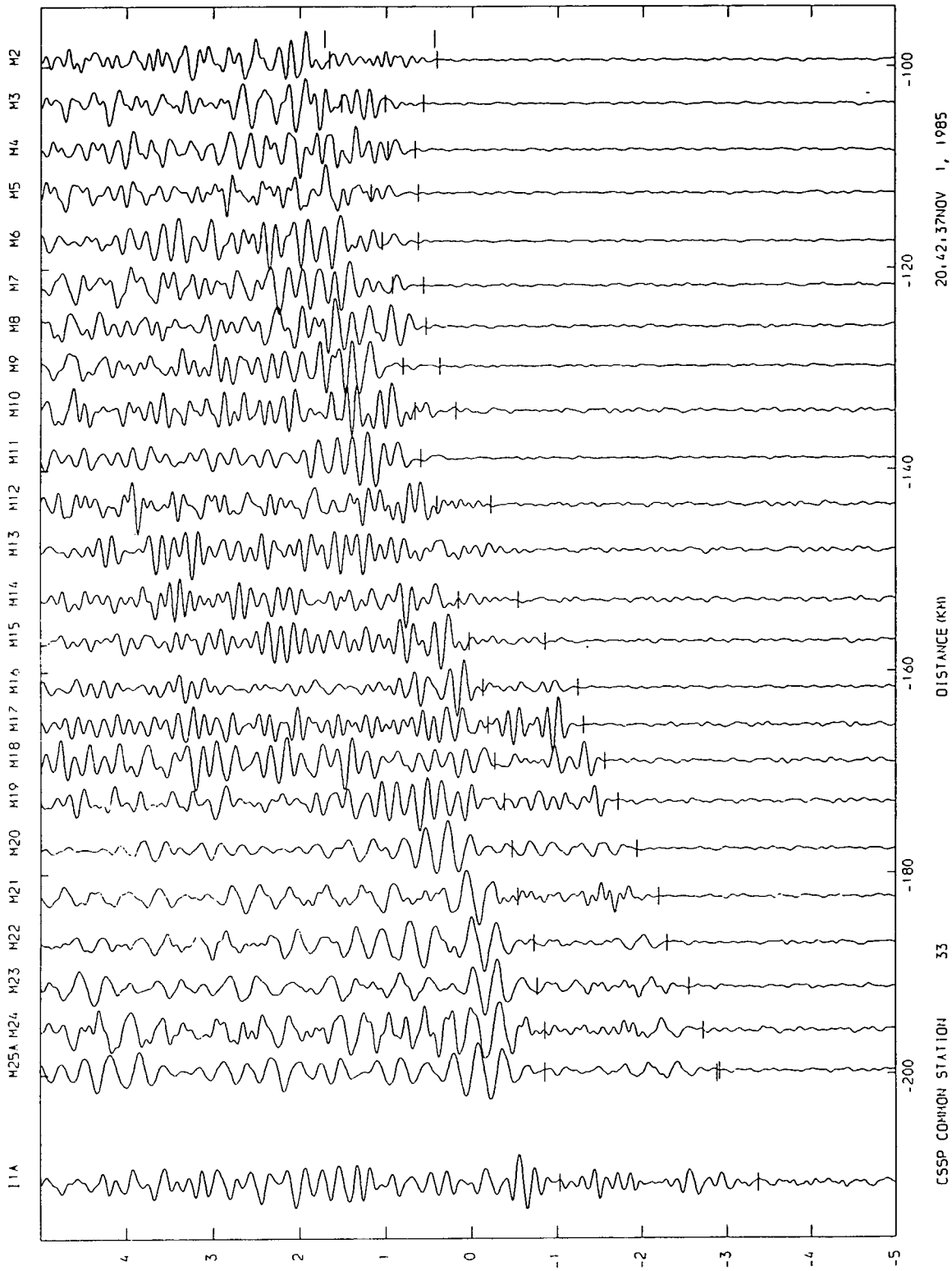
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



20:44:43NOV 1, 1985

CSSP COMMON STATION 32

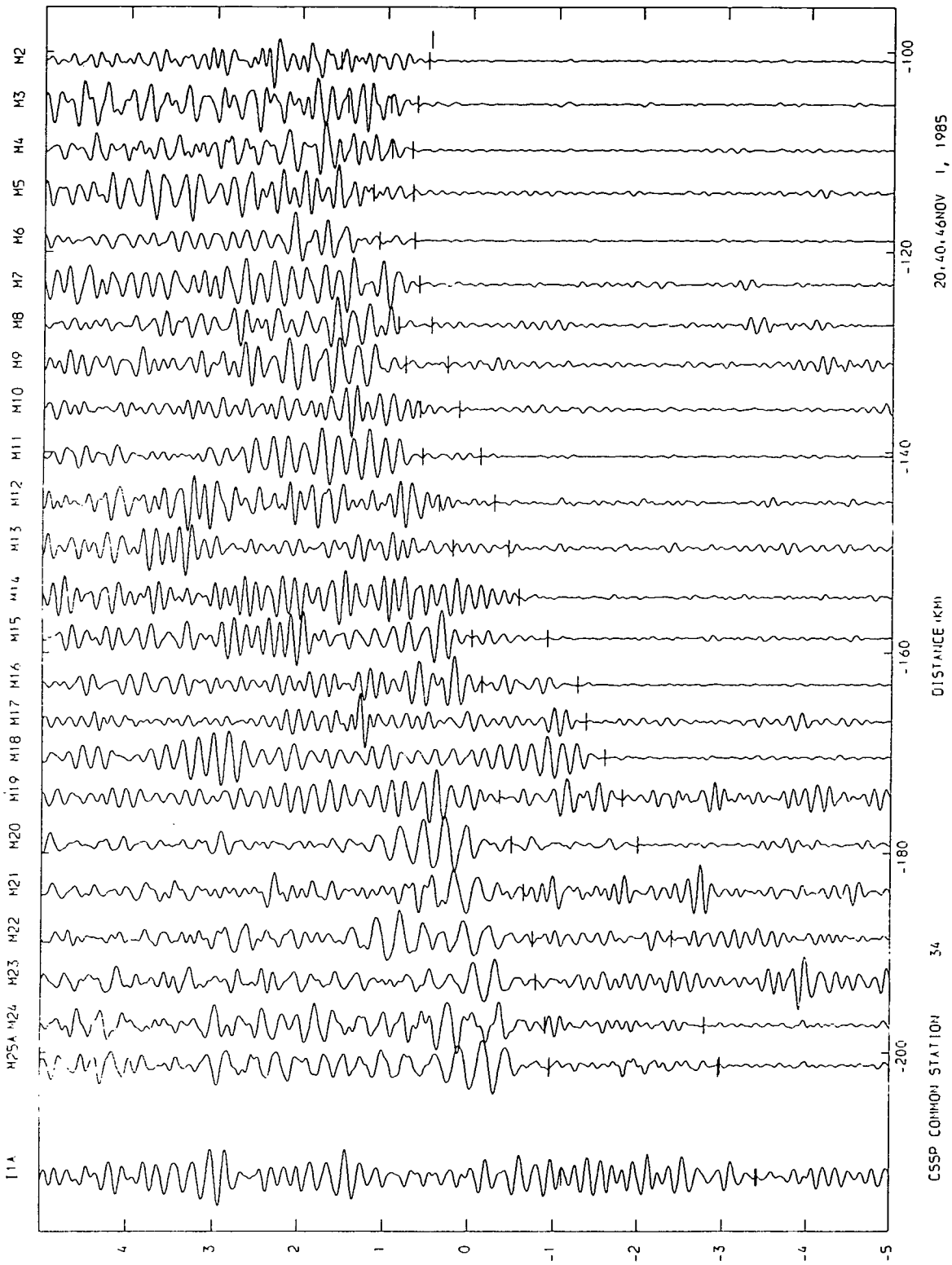
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



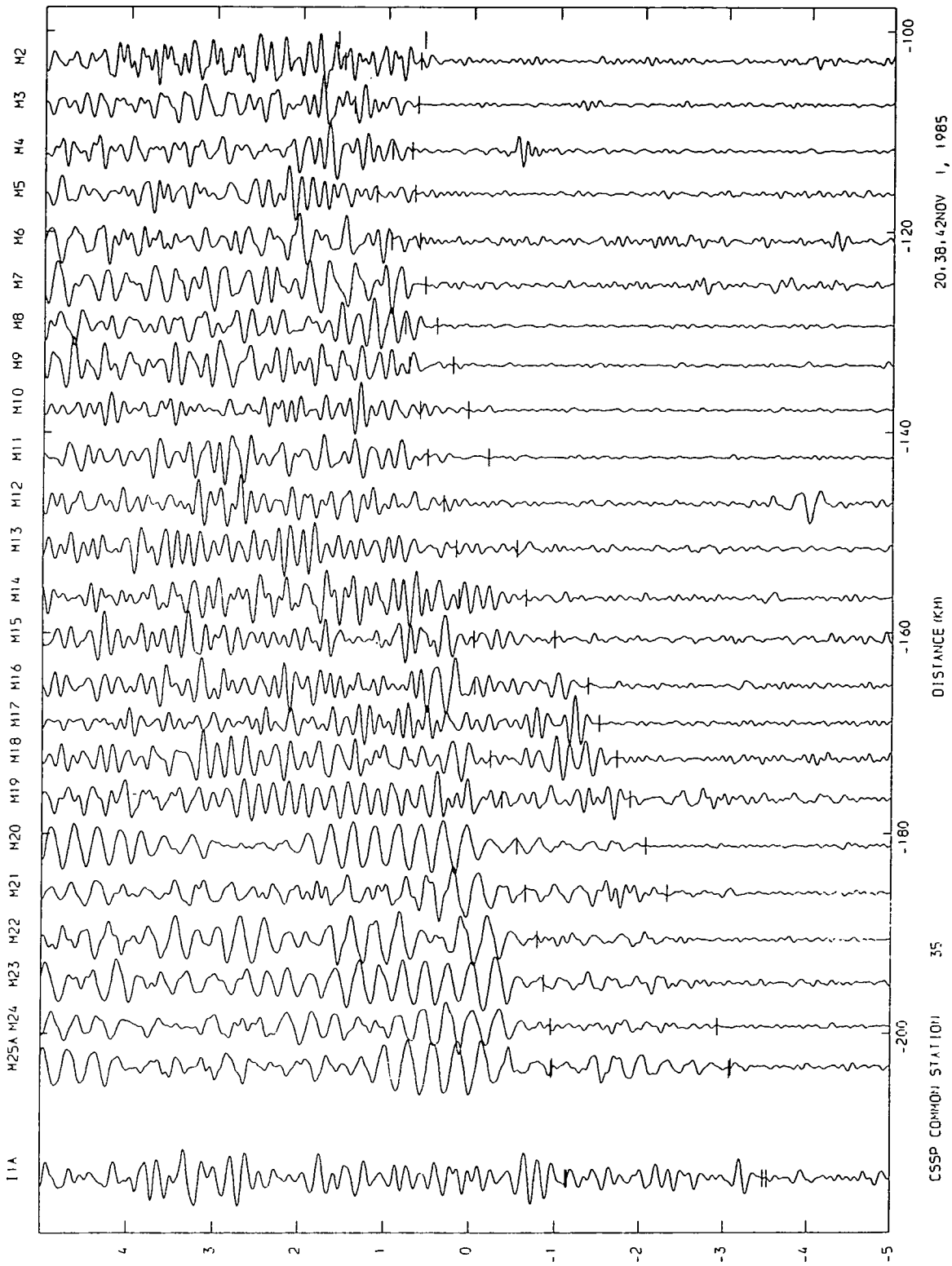
20:42:37NOV 1, 1985

CSSP COMMON STATION 33

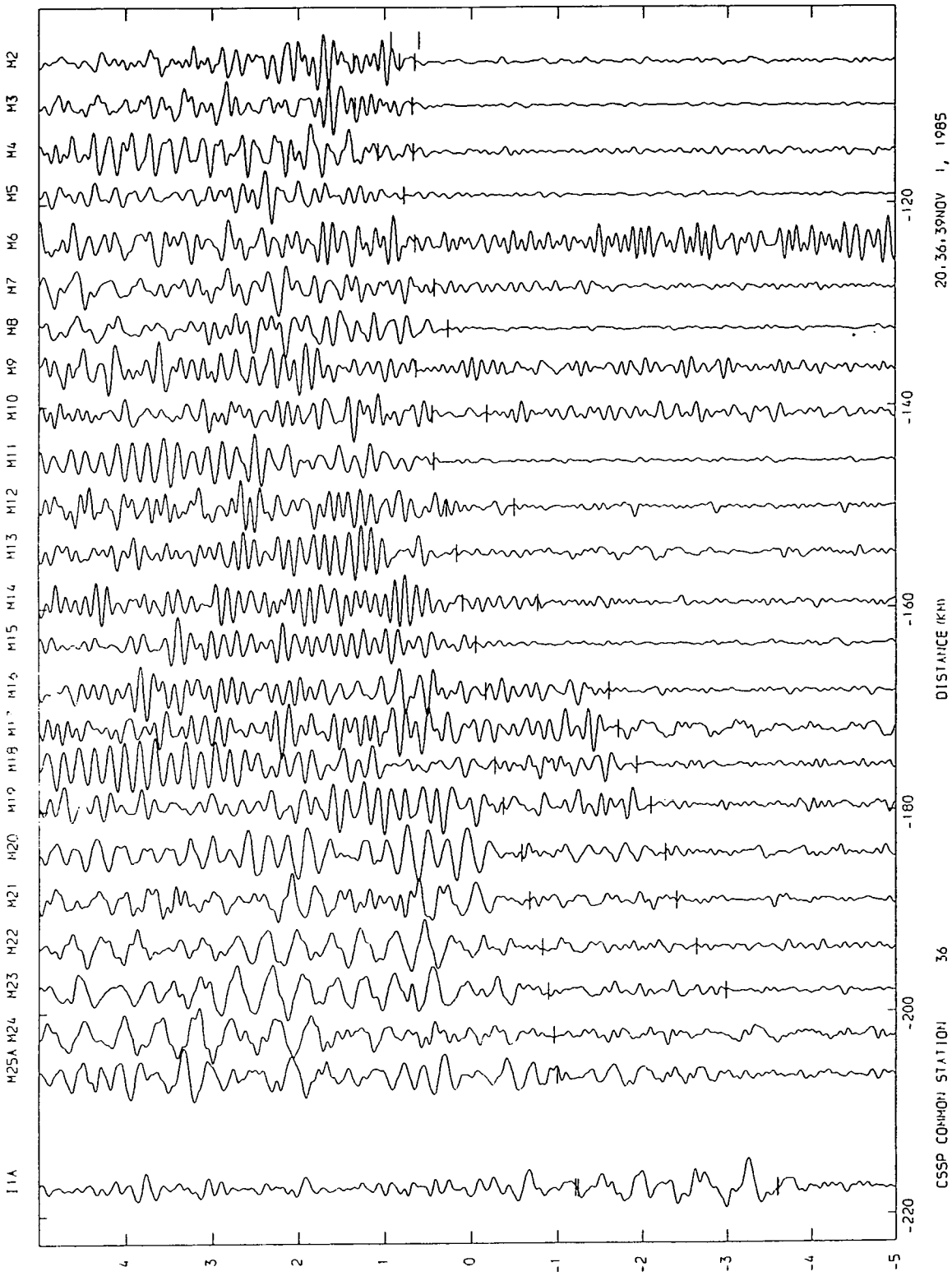
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



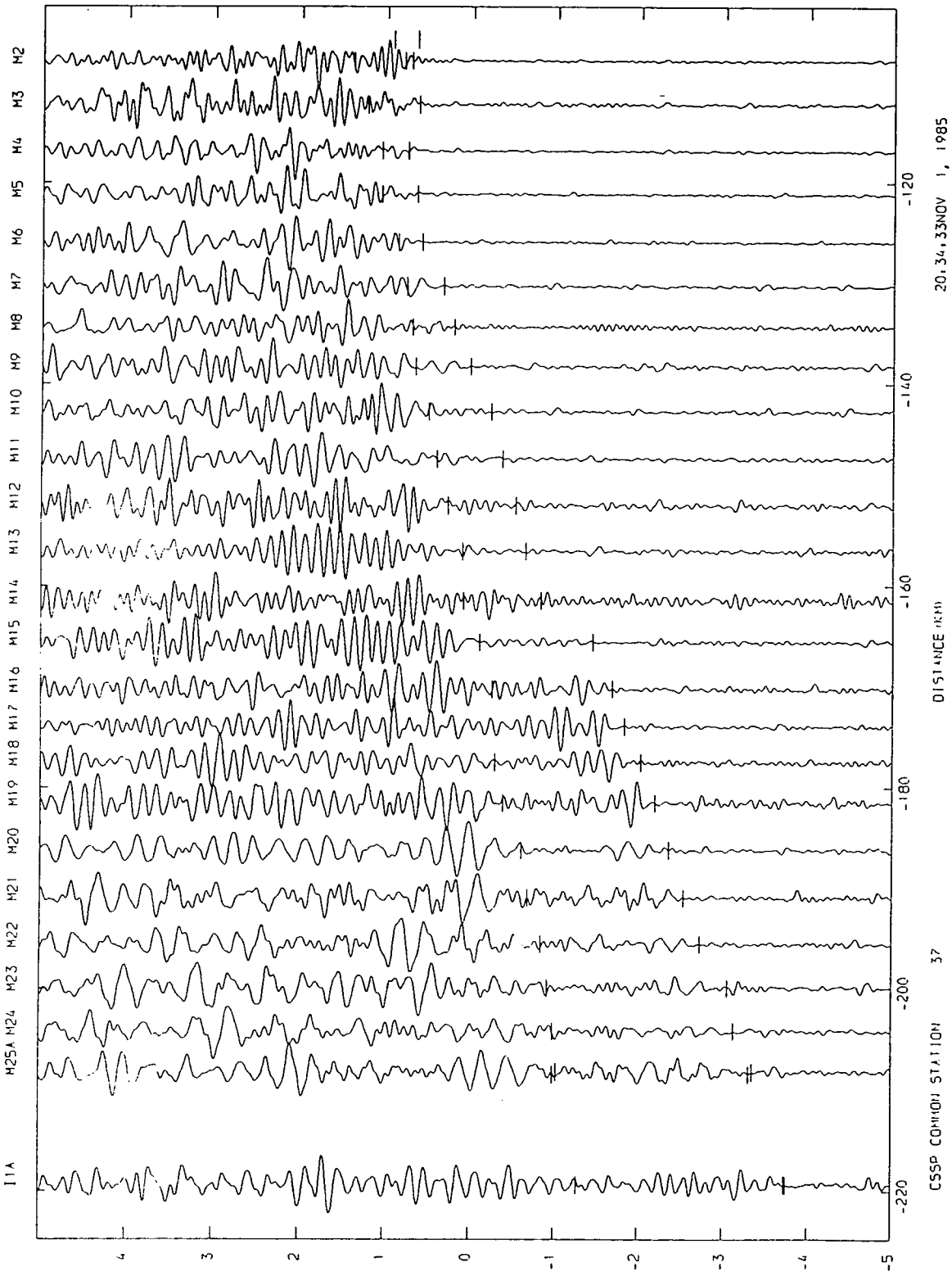
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



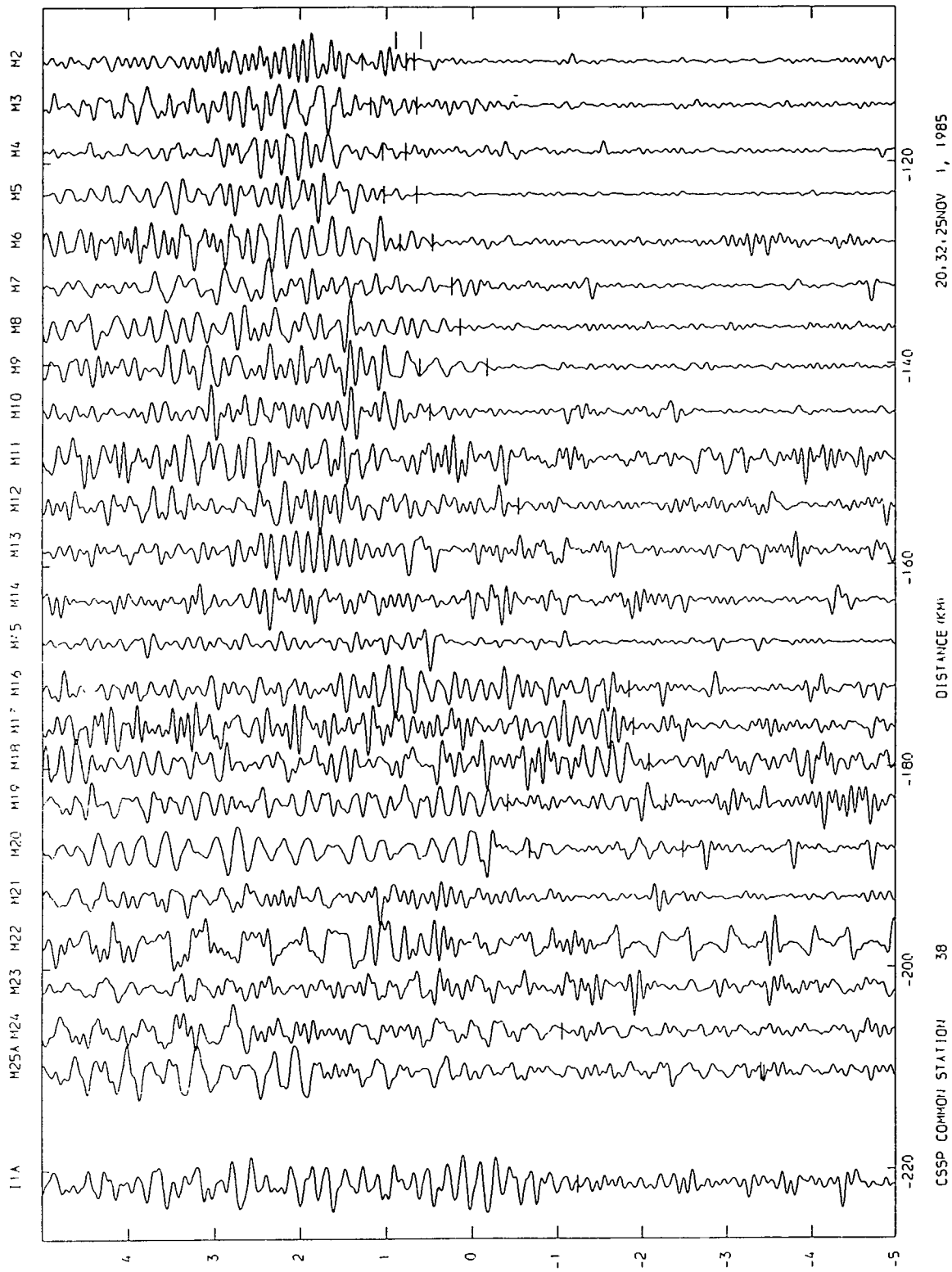
20:36:39NOV 1, 1985

CSSP COMMON STATION 36

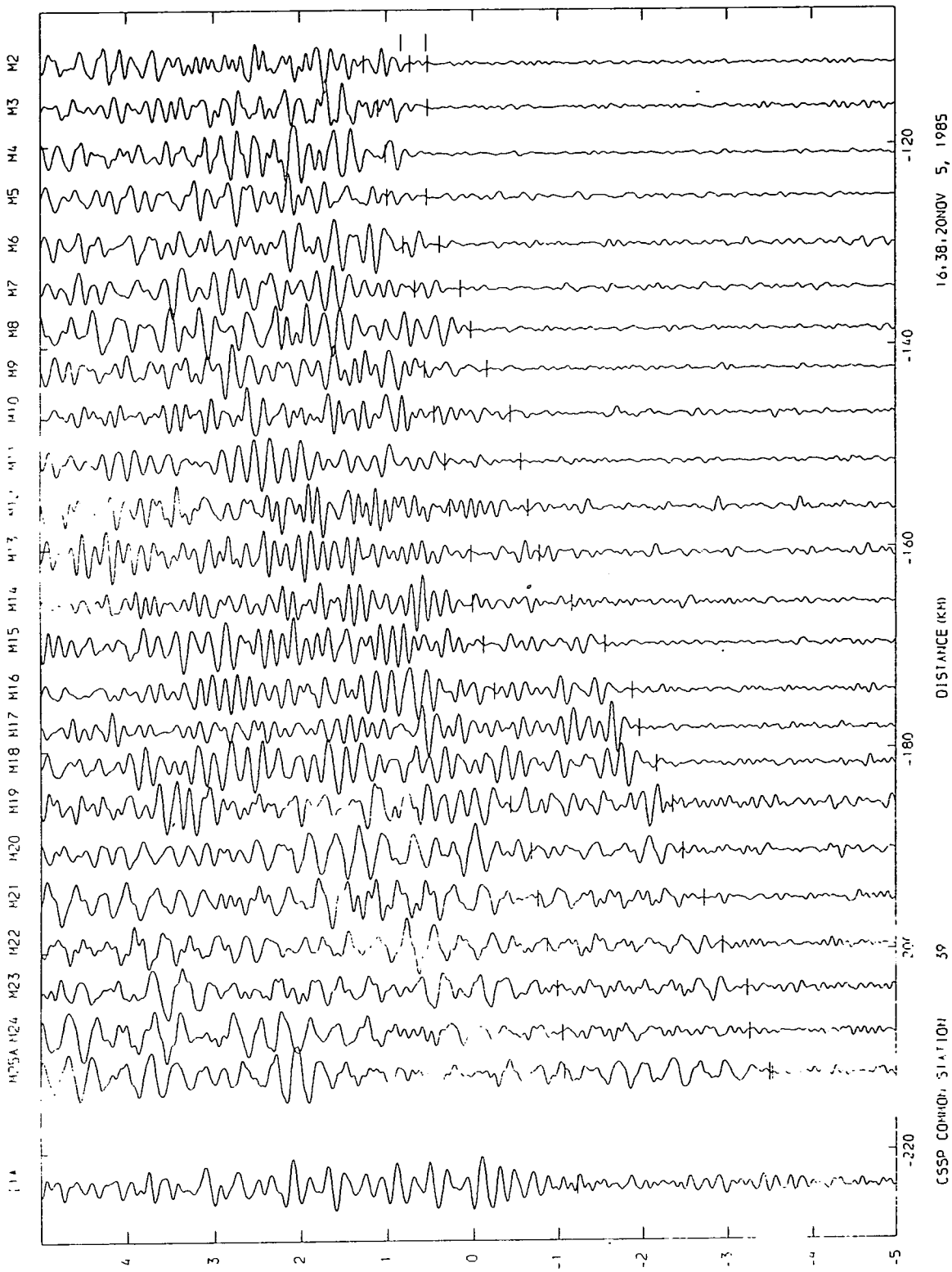
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



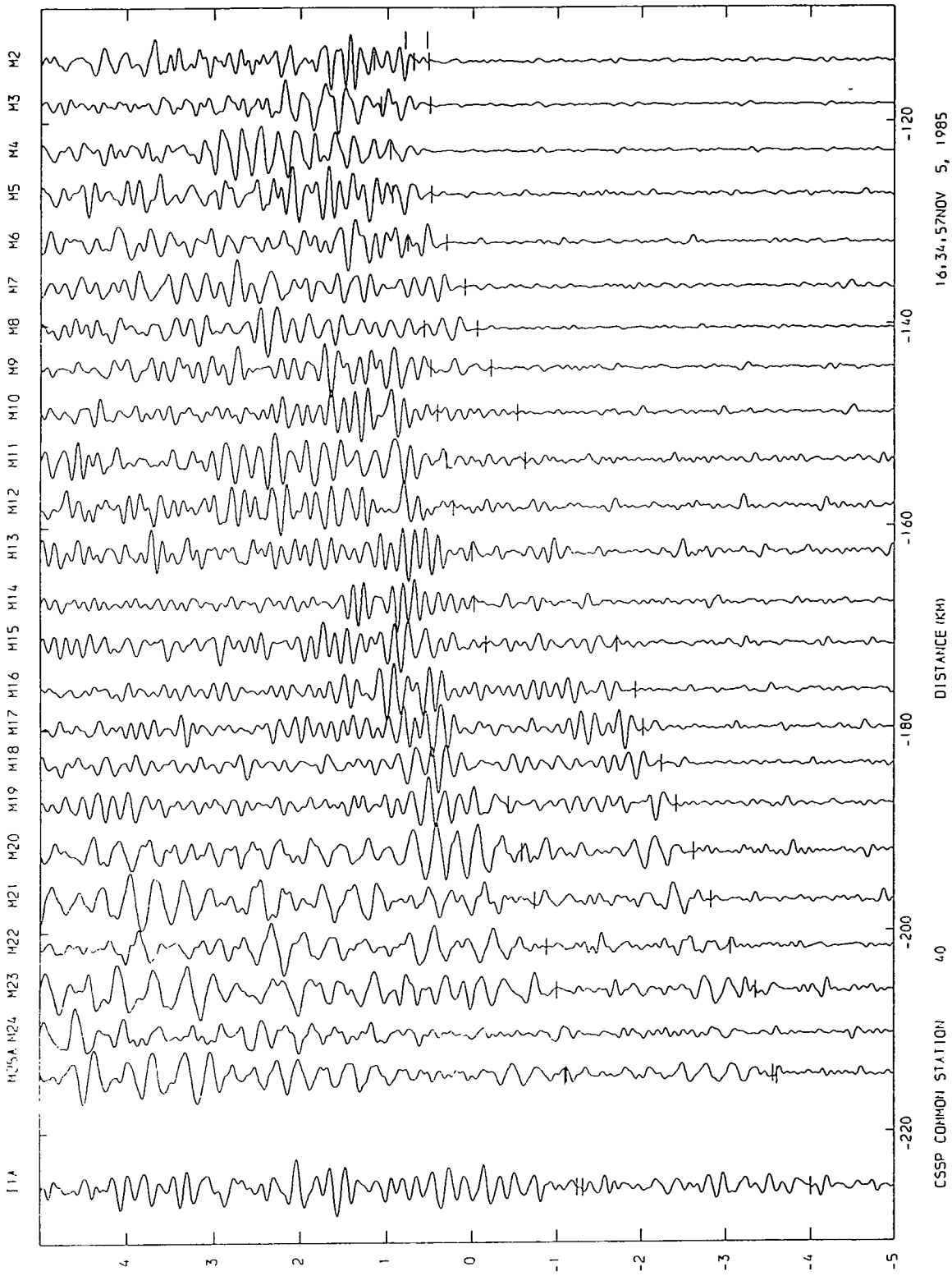
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



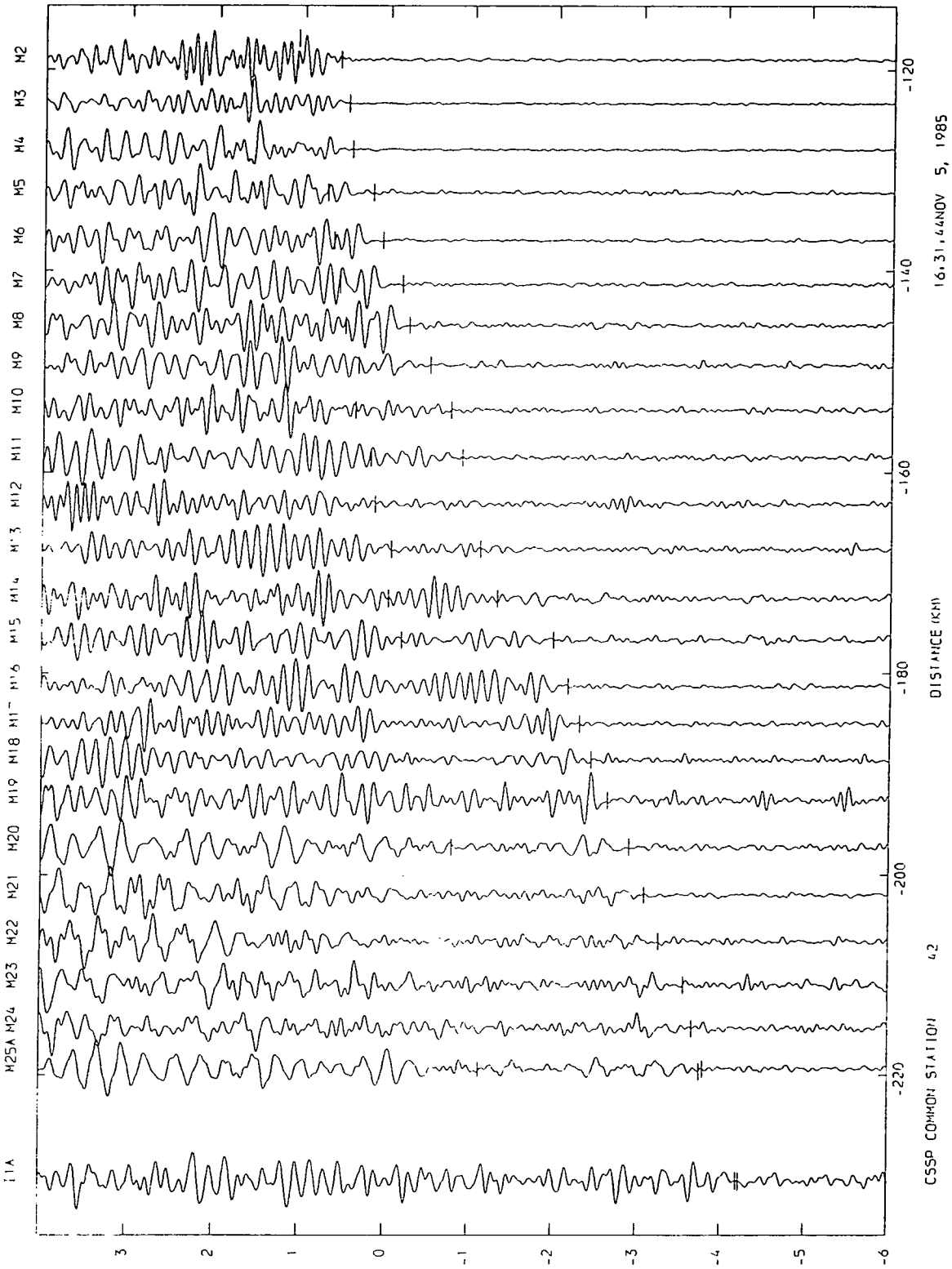
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



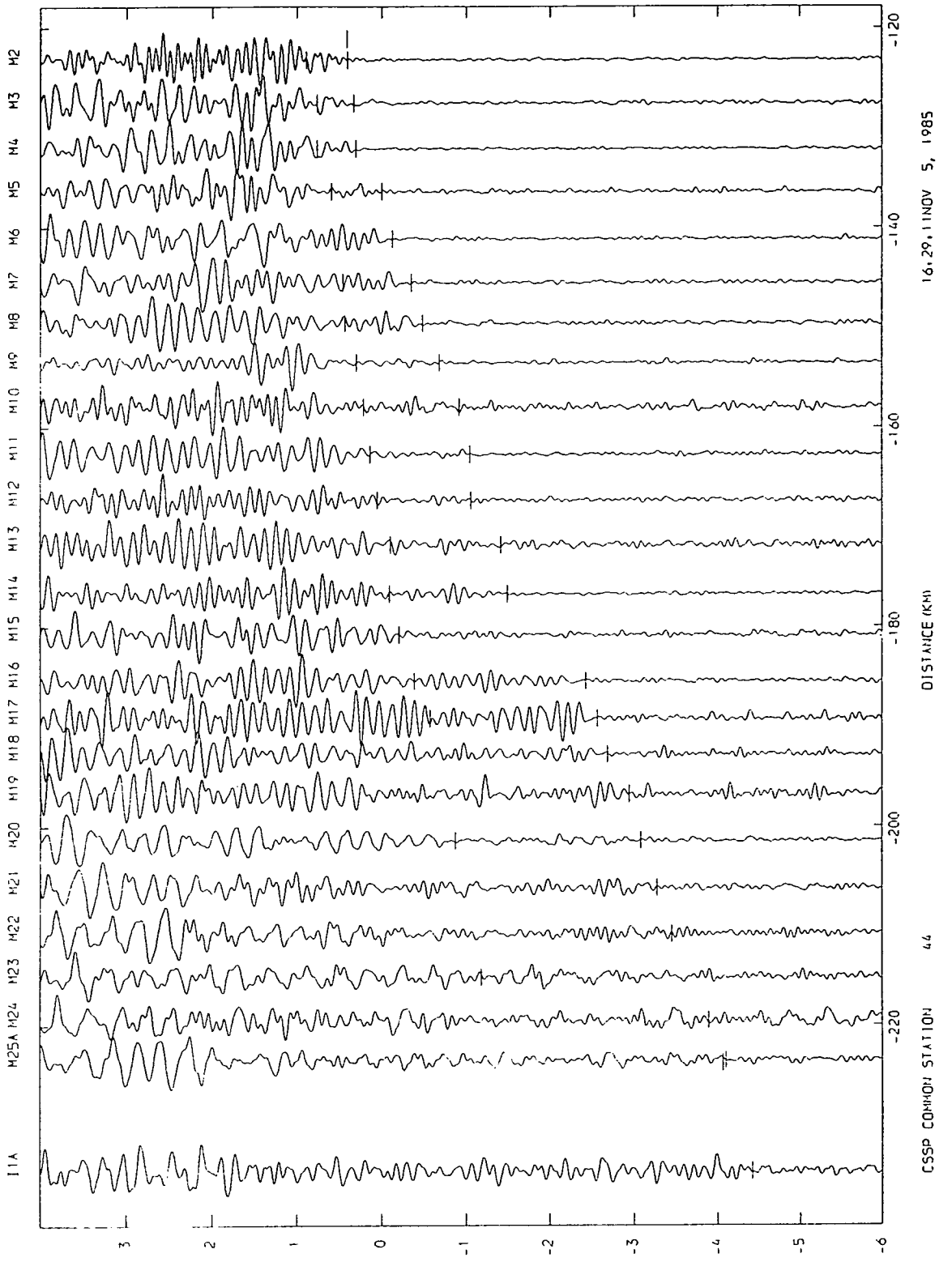
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



16.31.44NOV 5, 1985

CSSP COMMON STATION 42

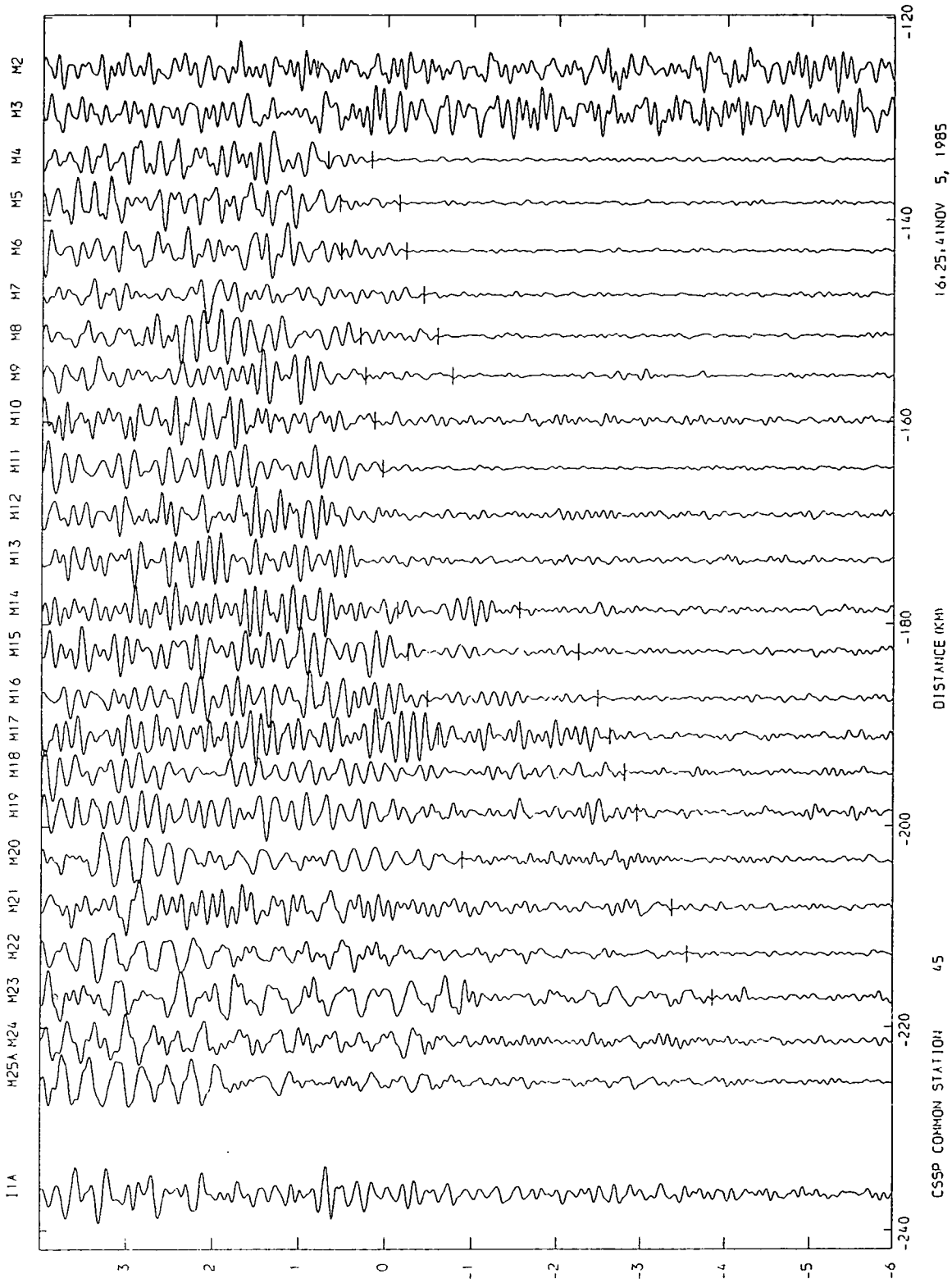
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



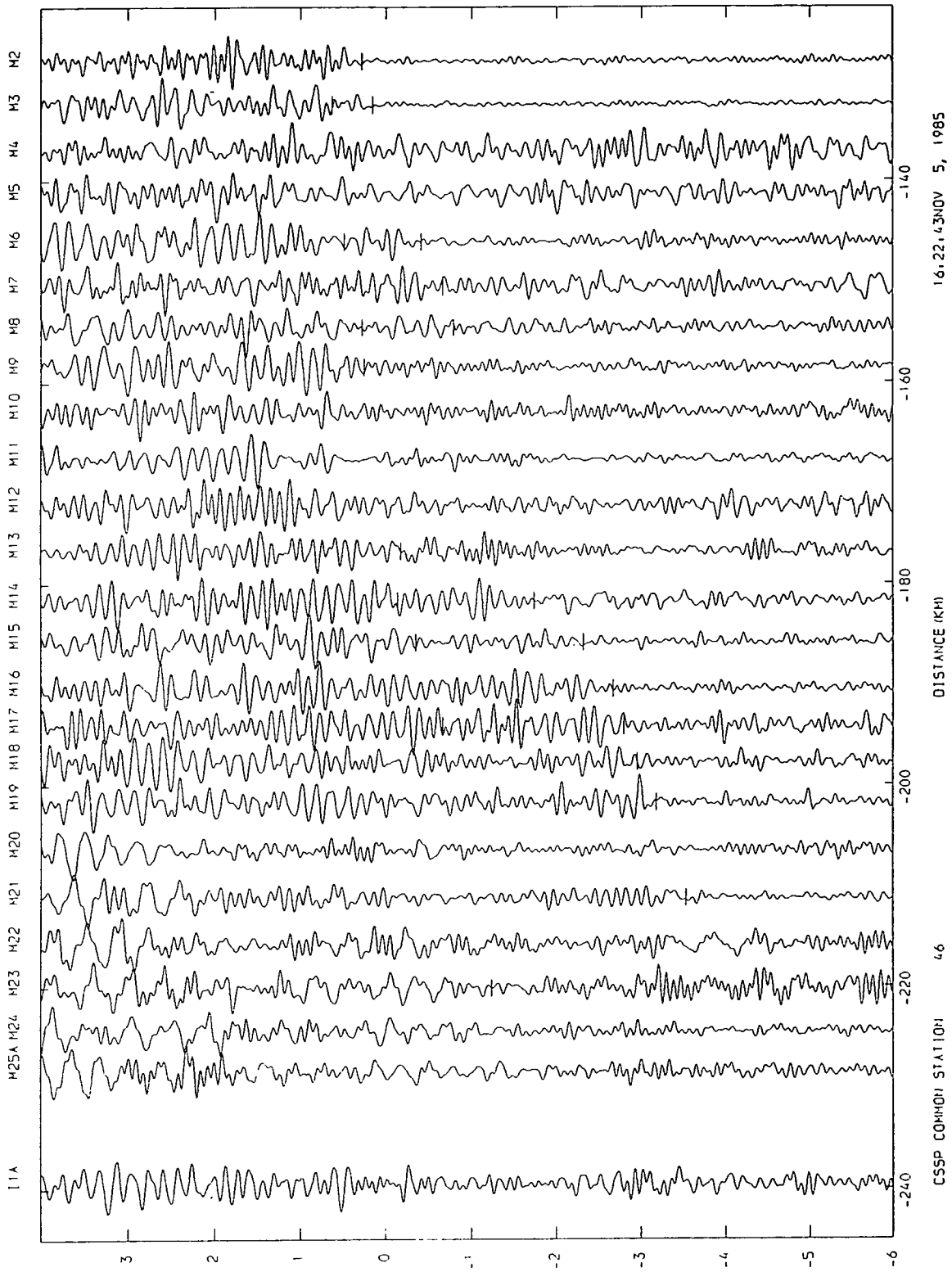
16:29,11NOV 5, 1985

CSSP COMMON STATION 44

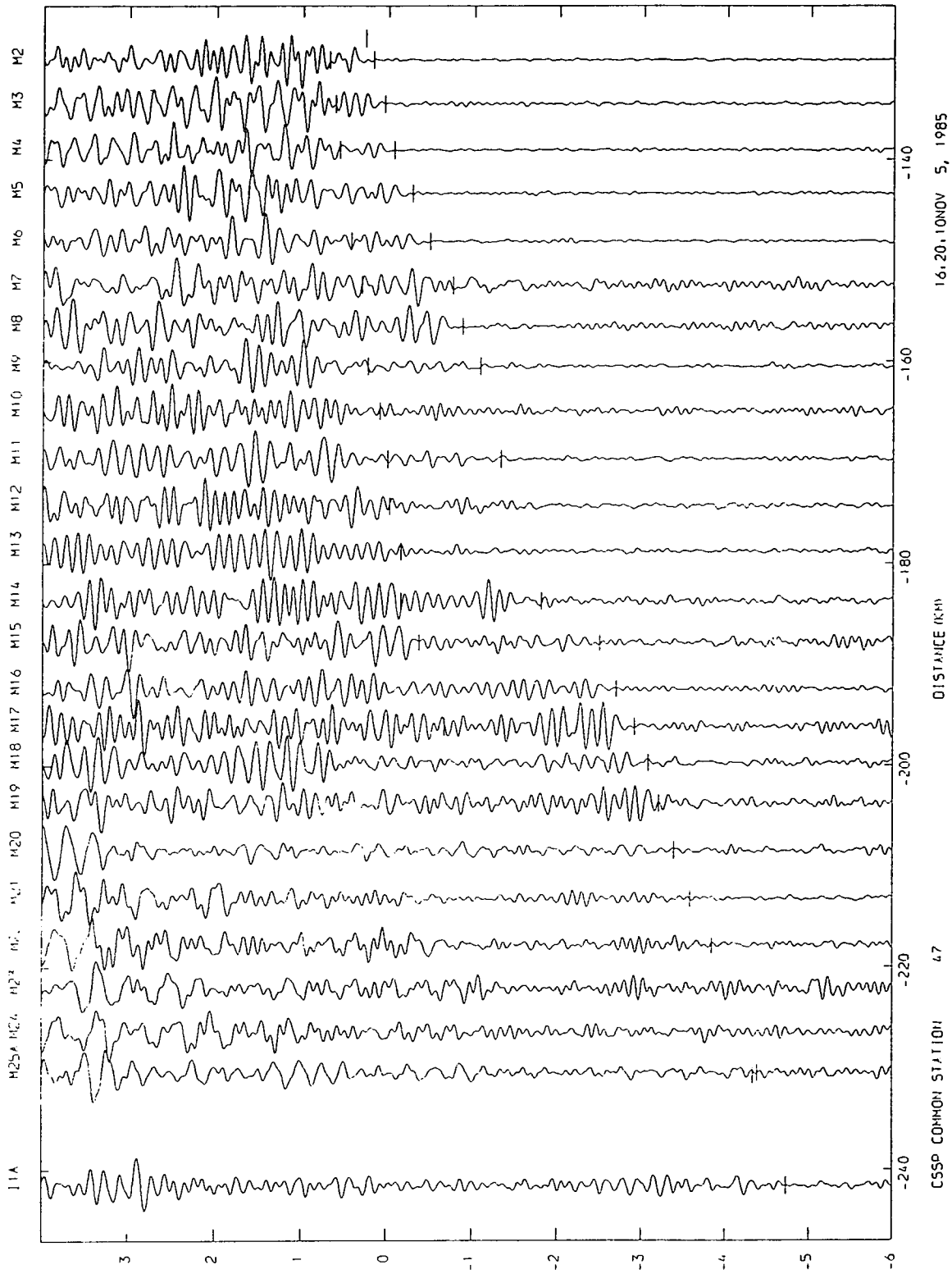
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



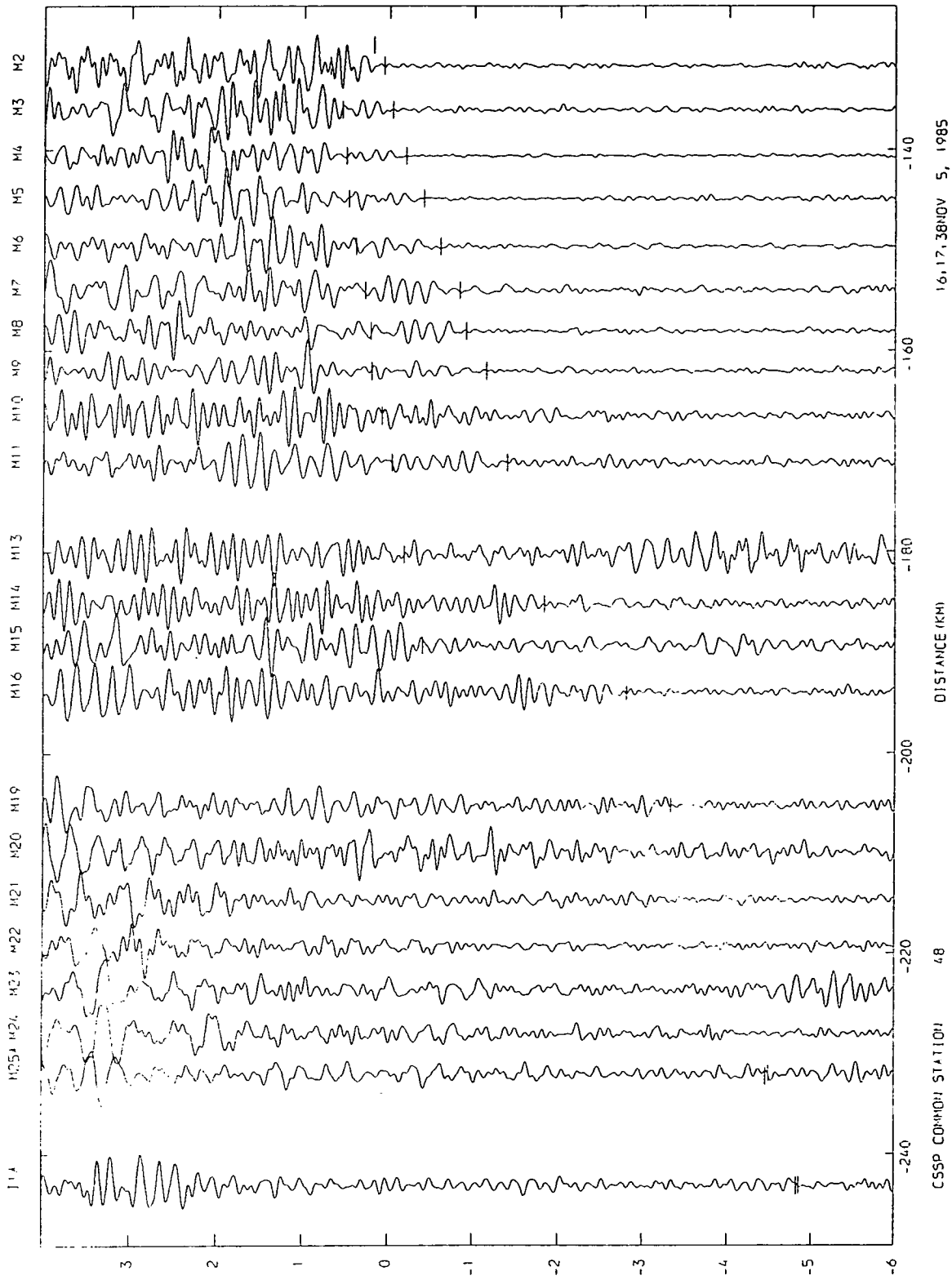
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



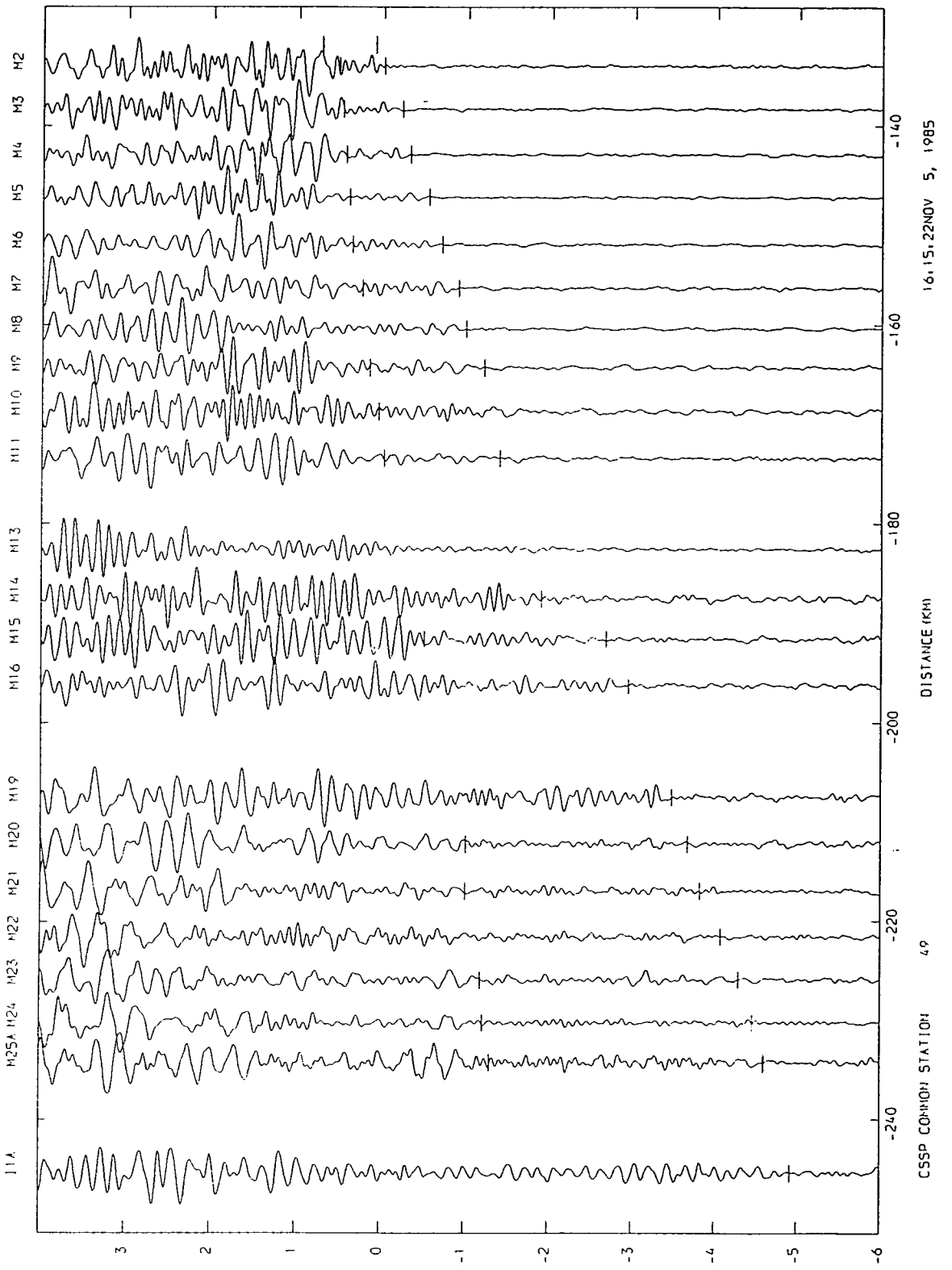
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



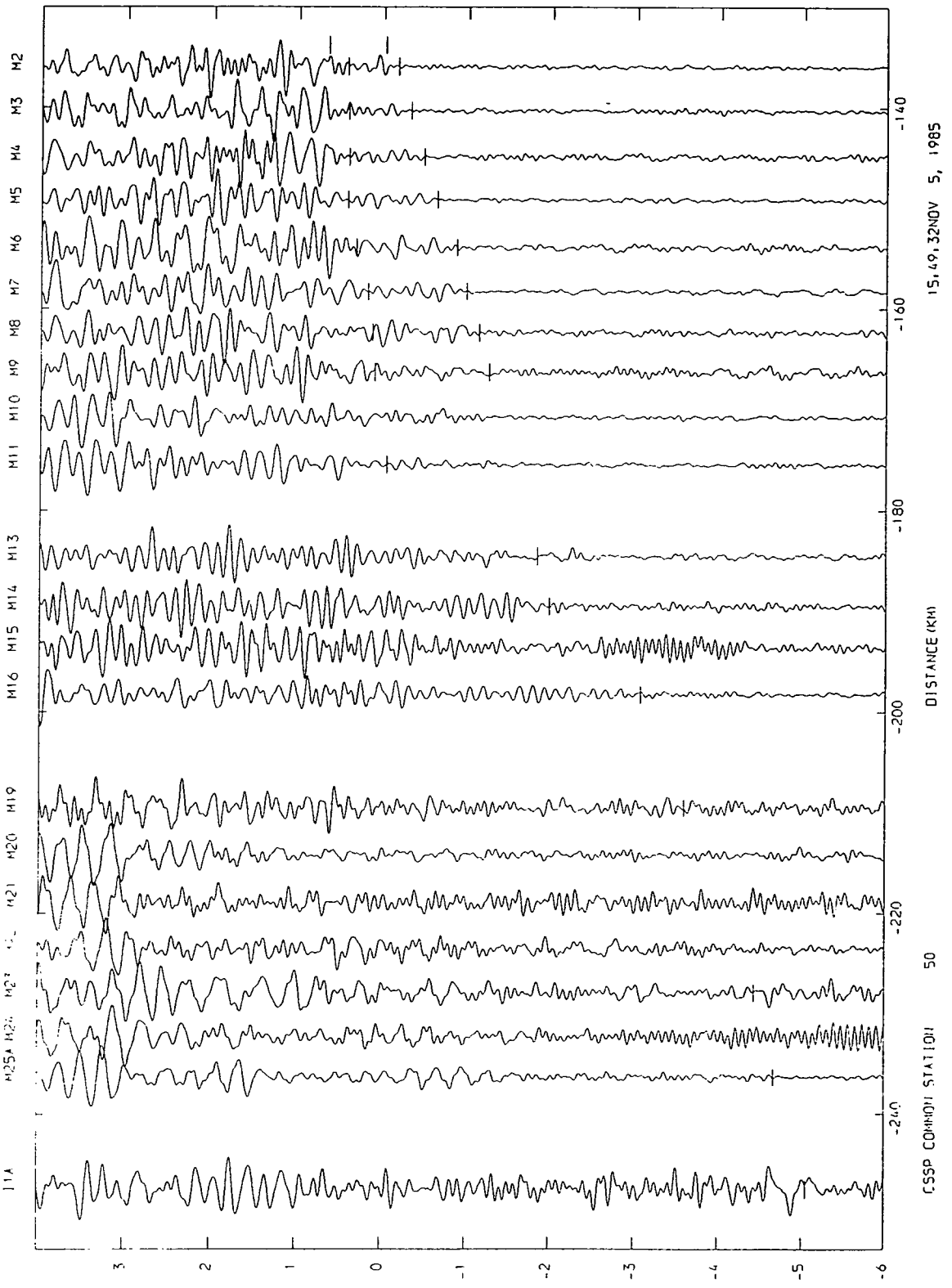
16,17,38NOV 5, 1985

CSSP COMMON STATION 48

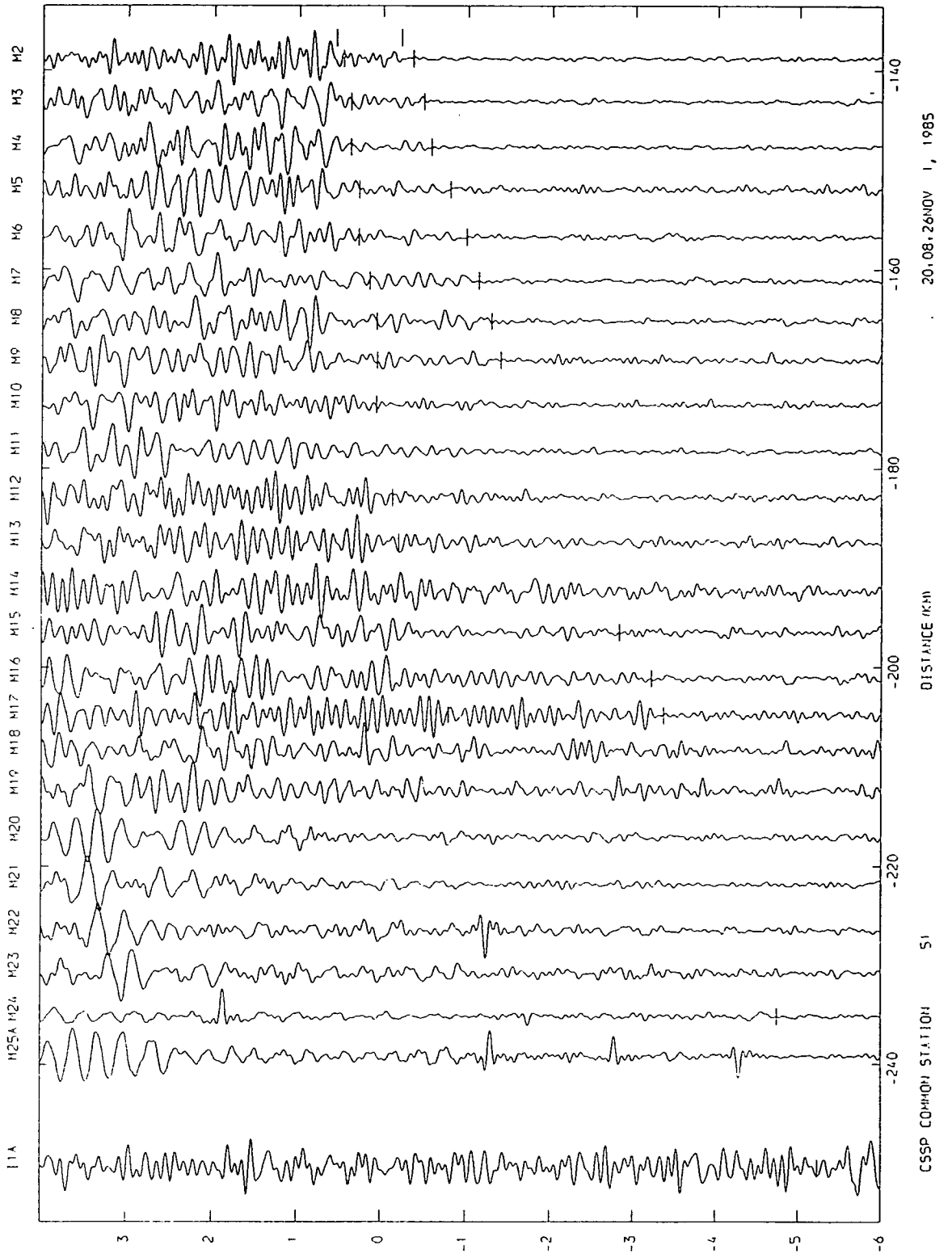
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



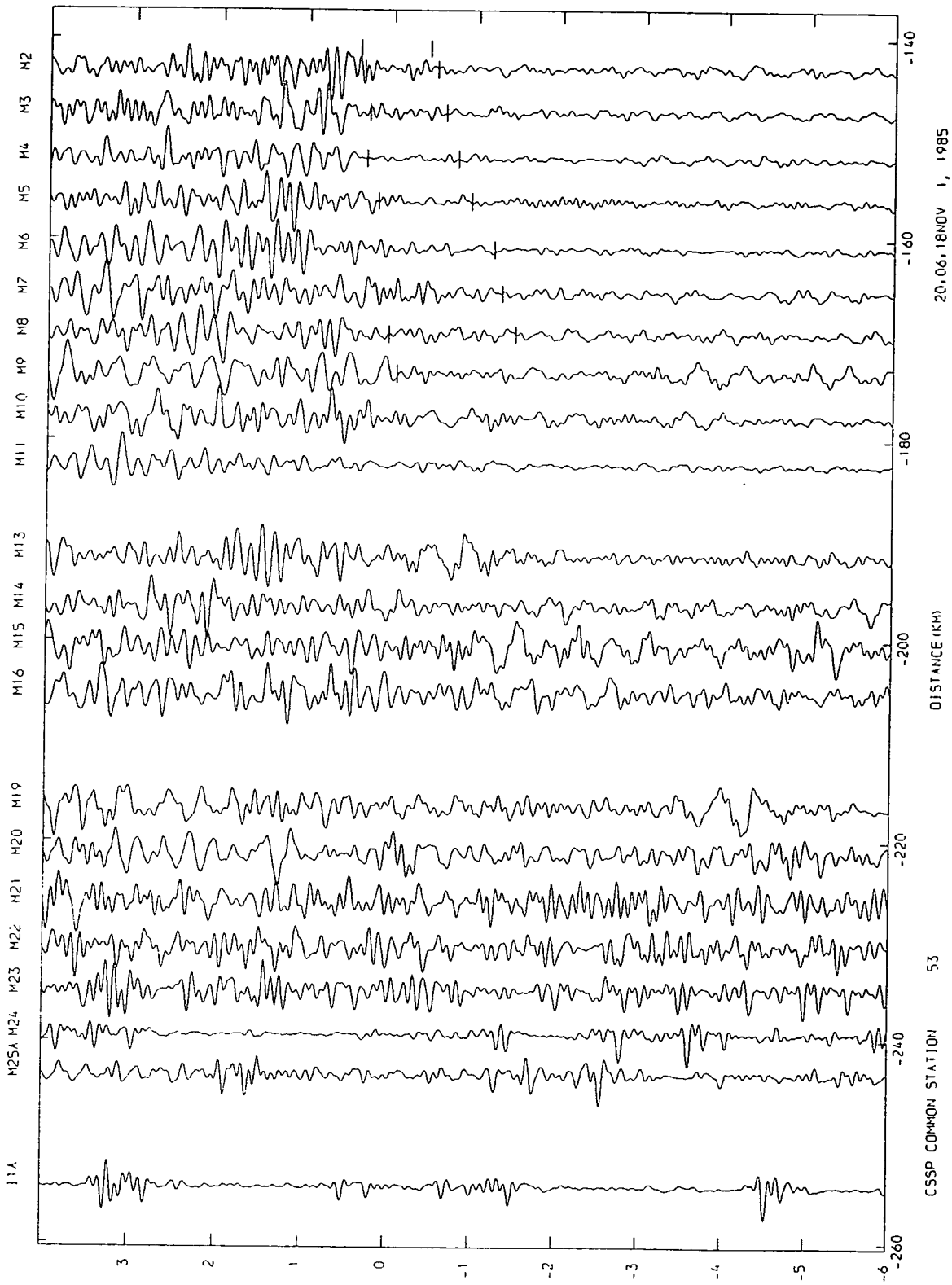
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



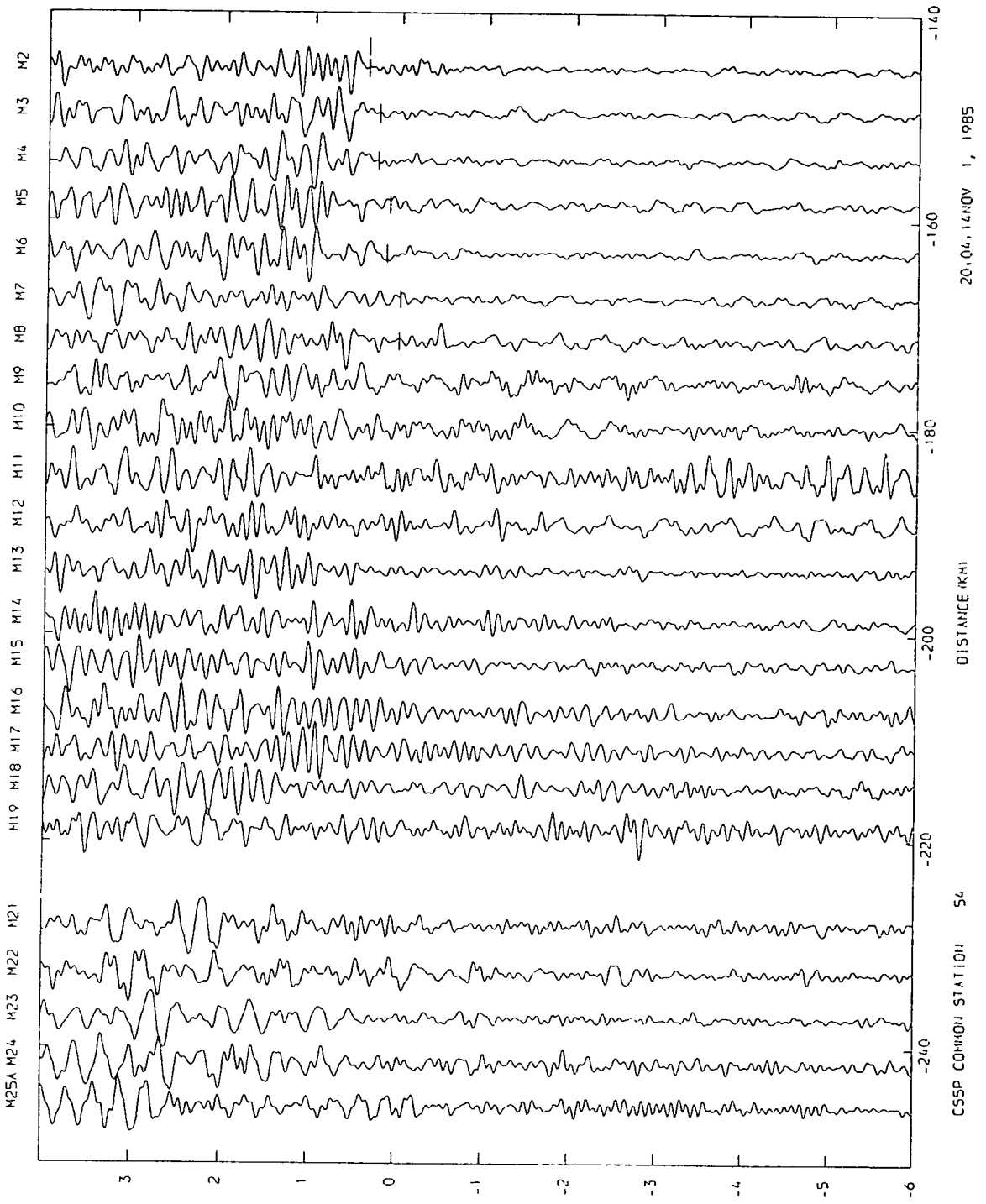
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



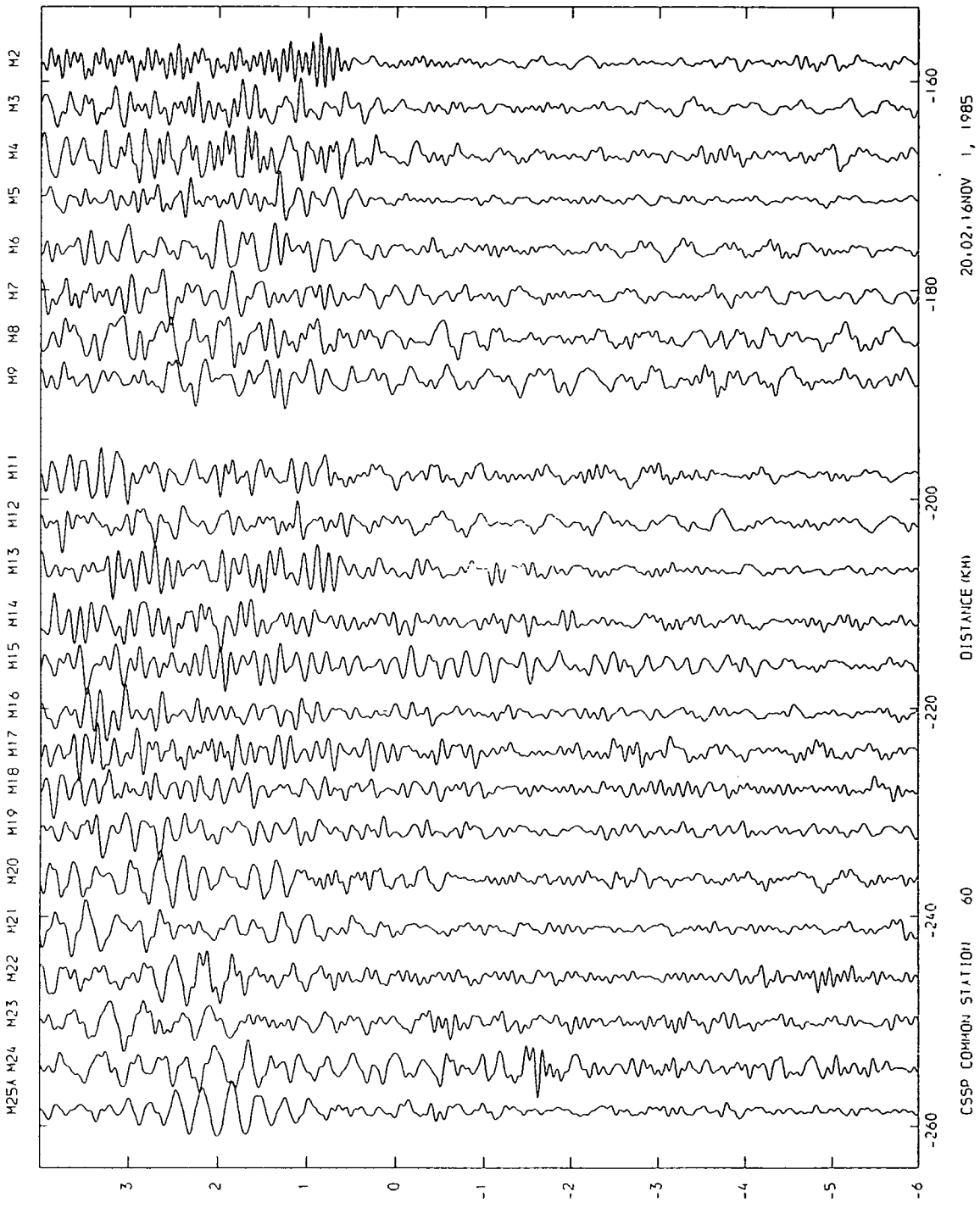
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



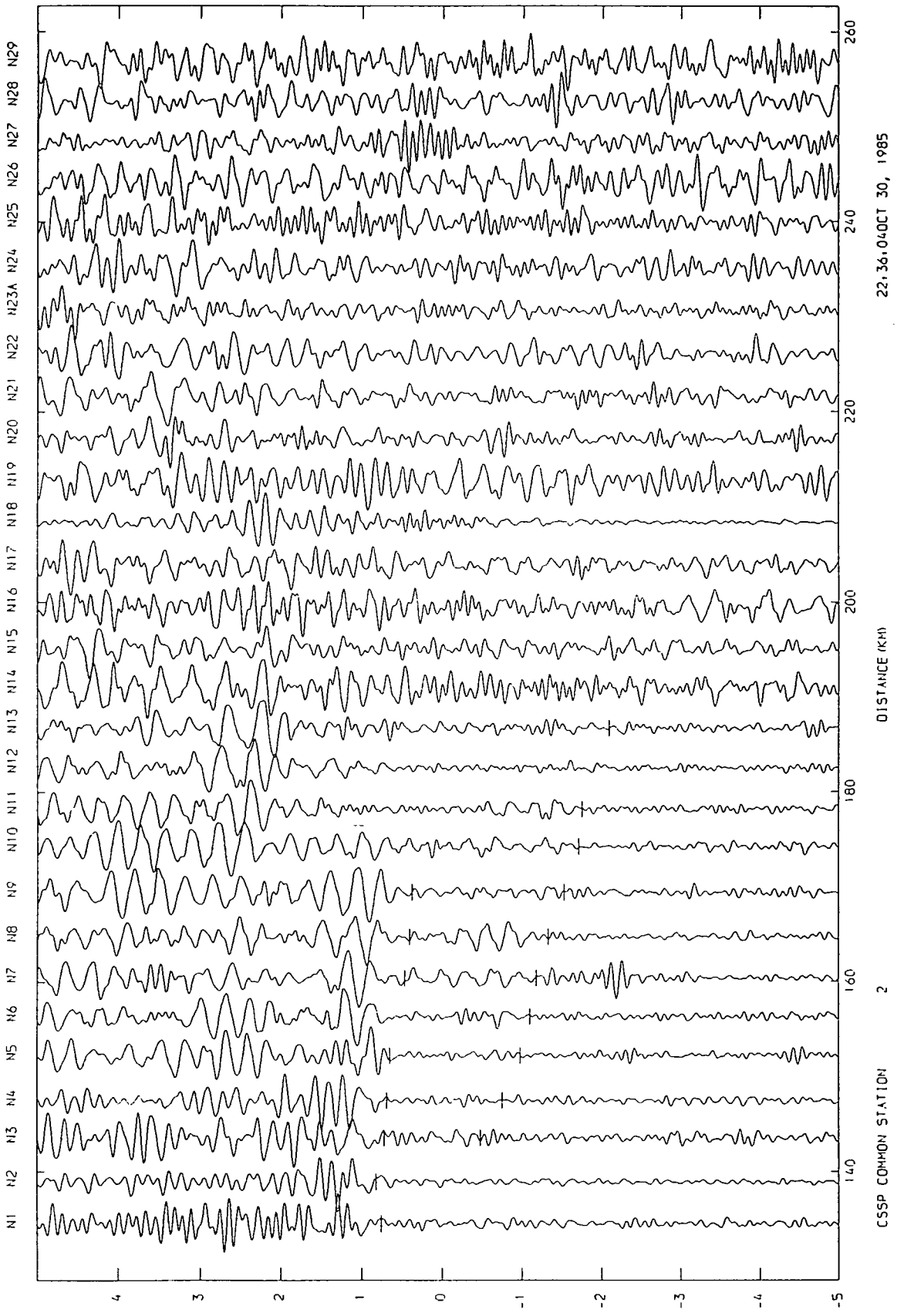
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



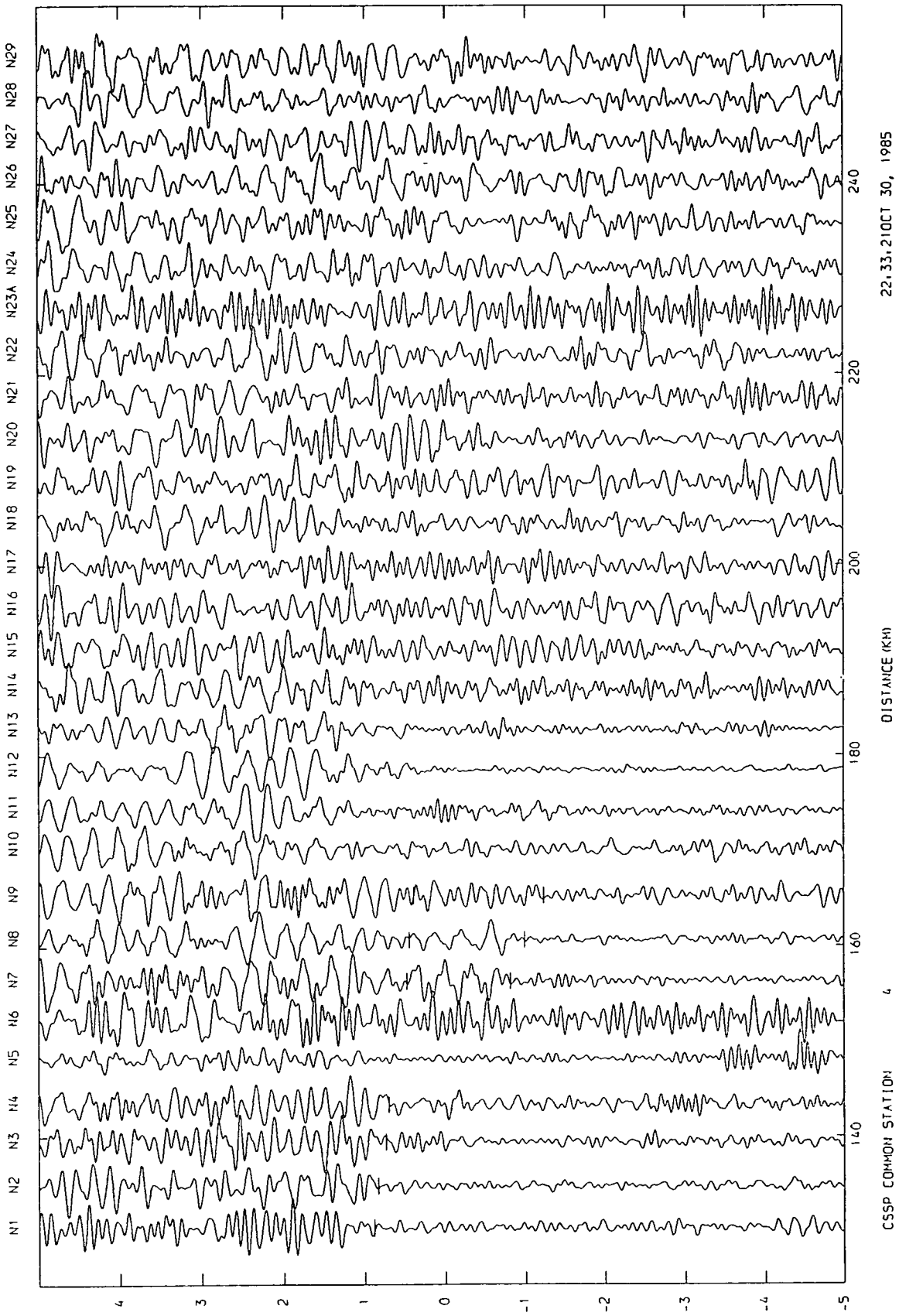
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



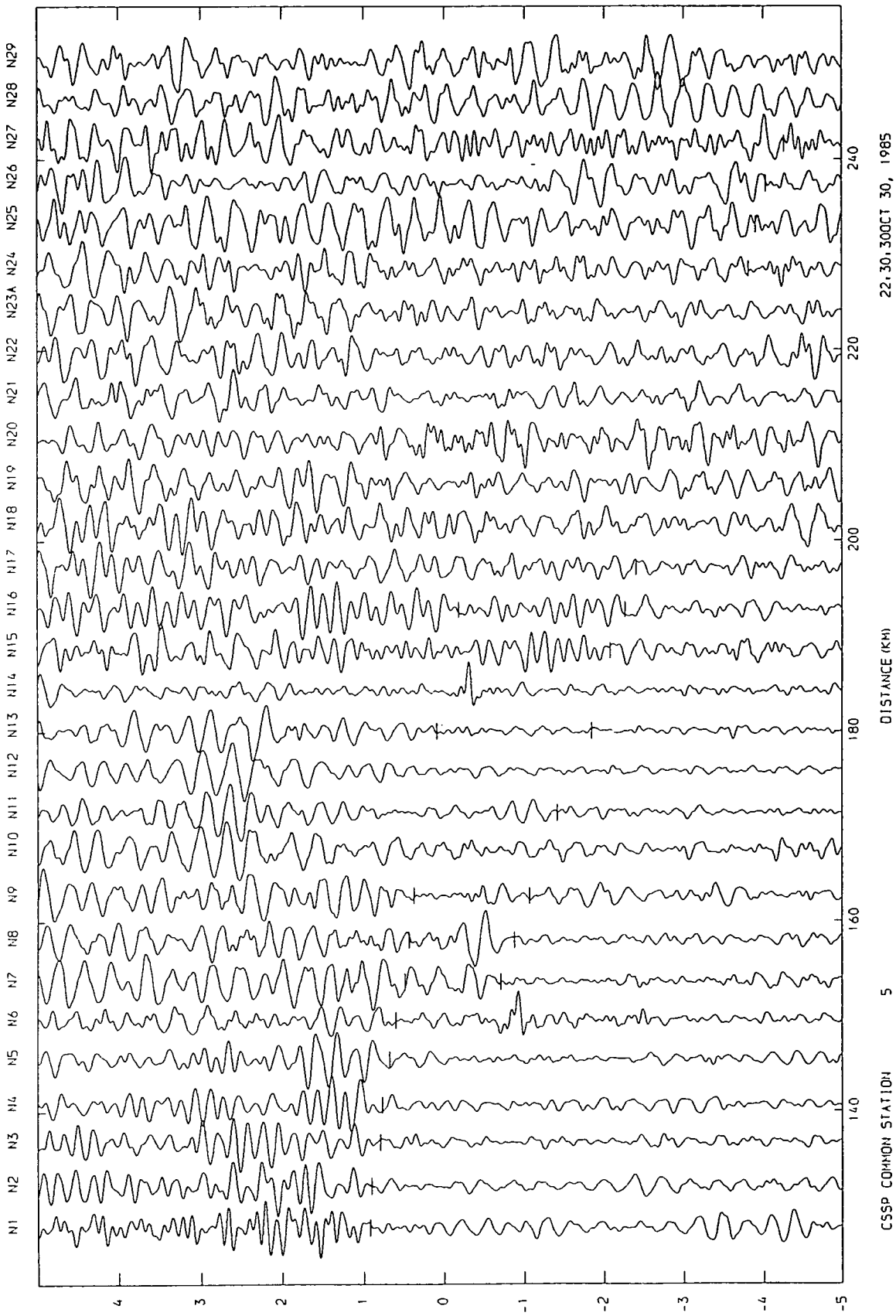
CSSP COMMON STATION 2

22:36:04OCT 30, 1985

FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



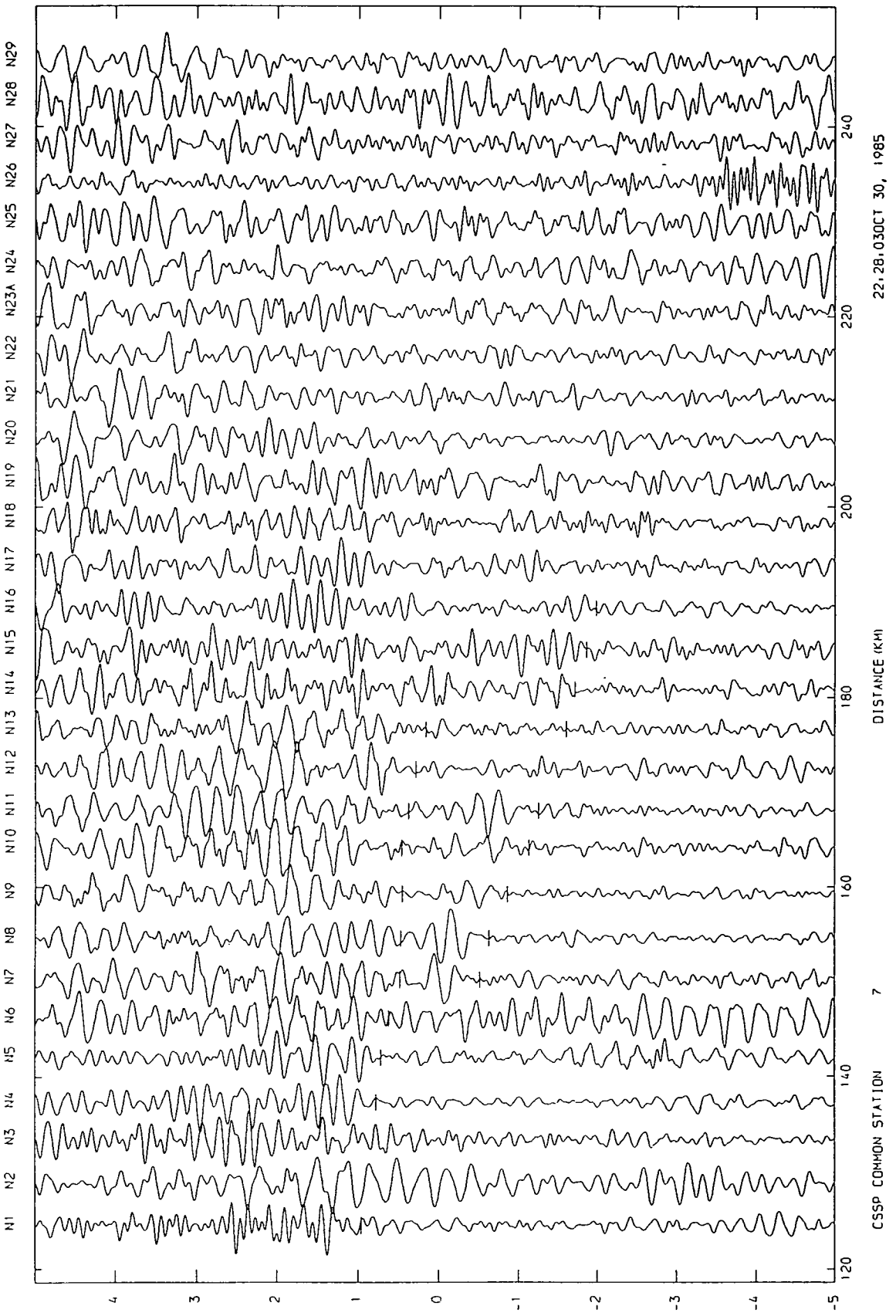
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



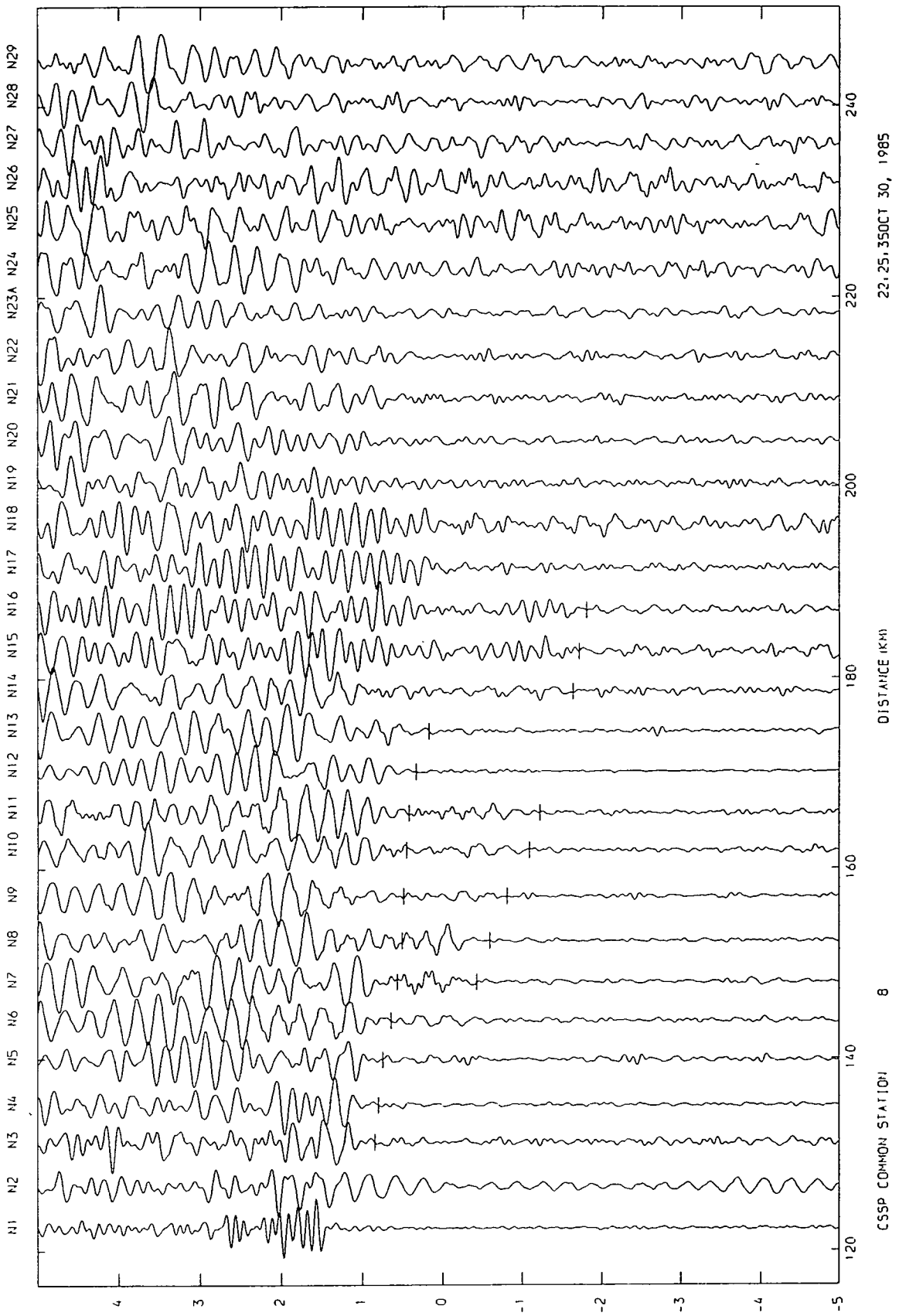
CSSP COMMON STATION 5

22:30:30OCT 30, 1985

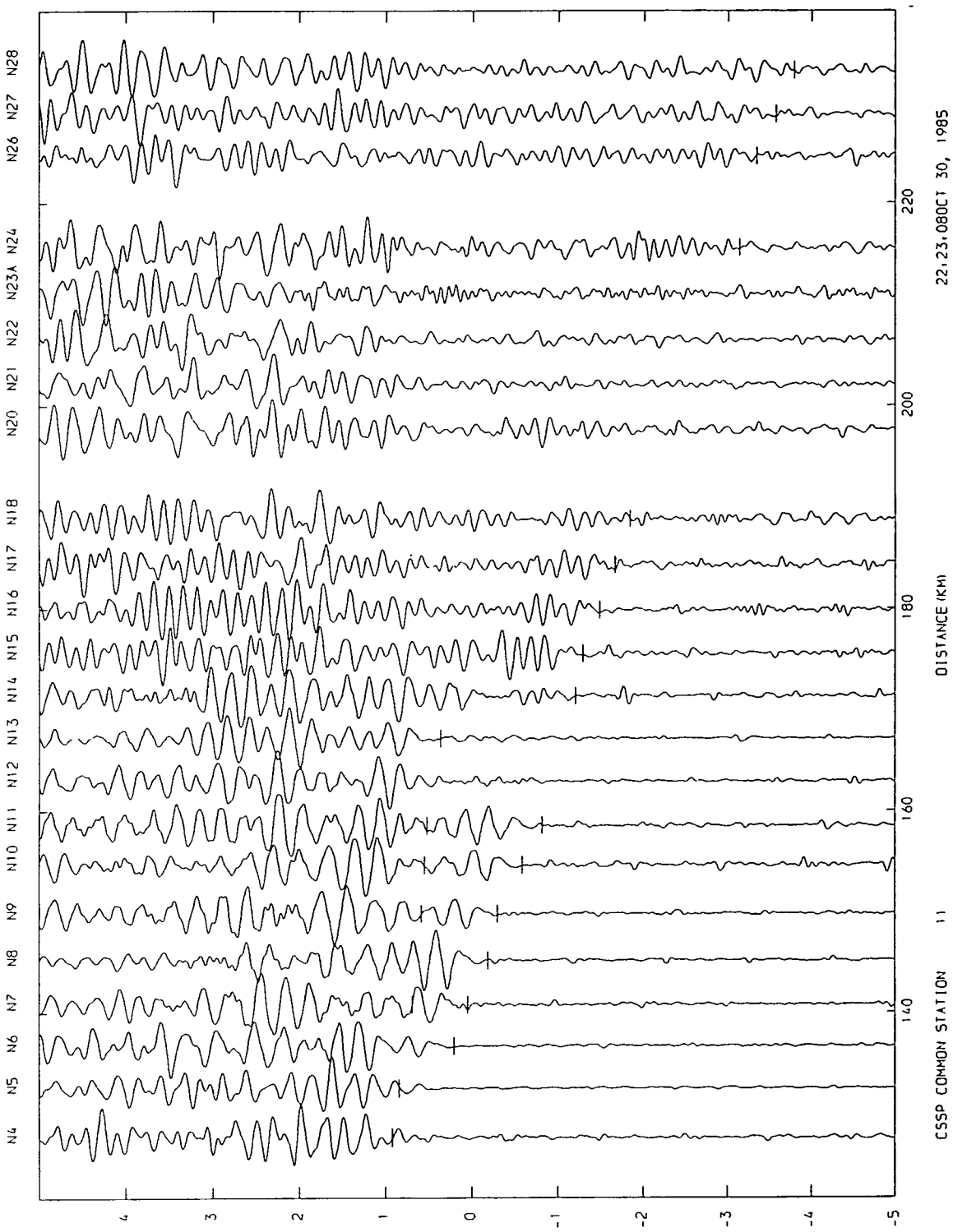
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



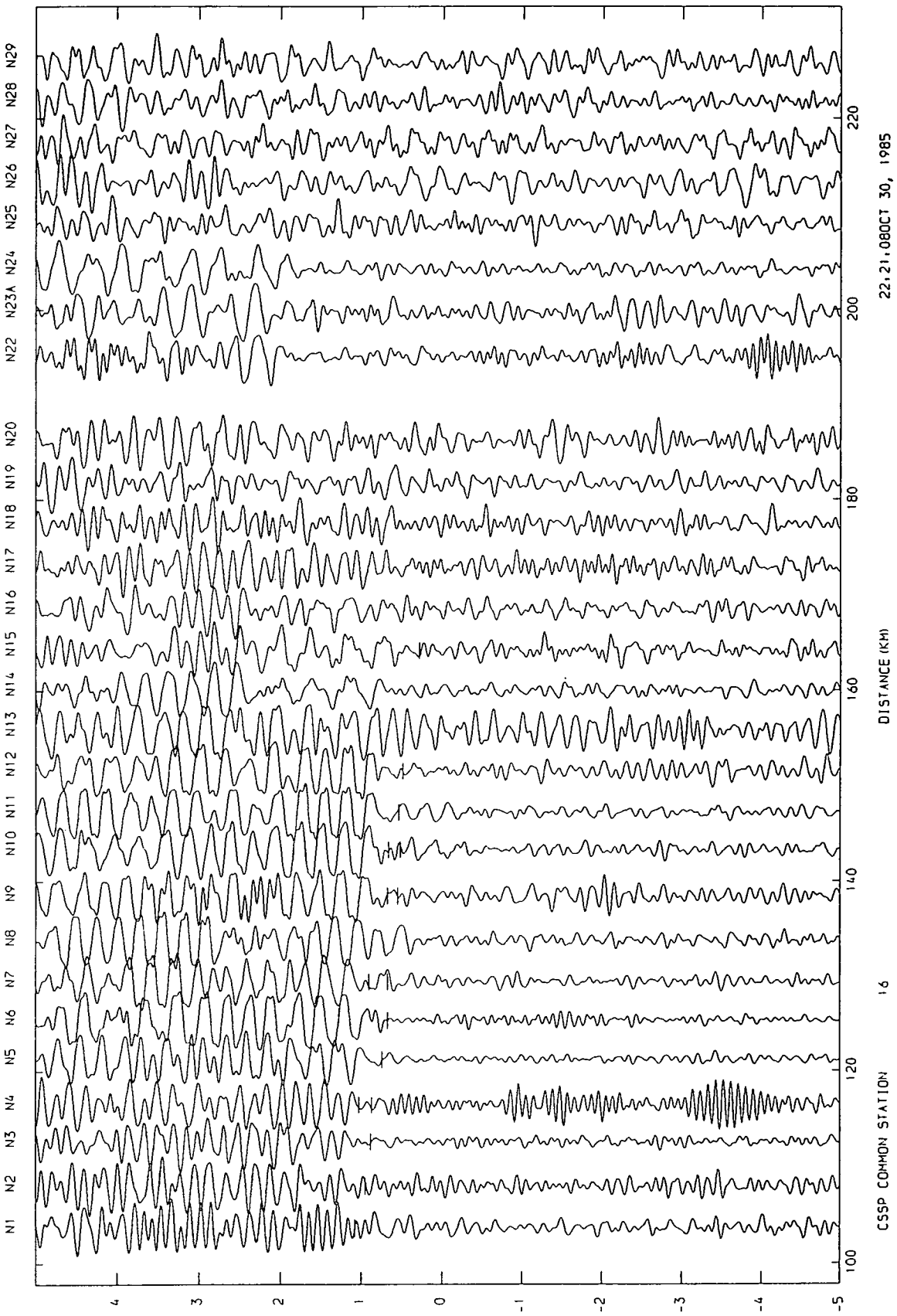
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



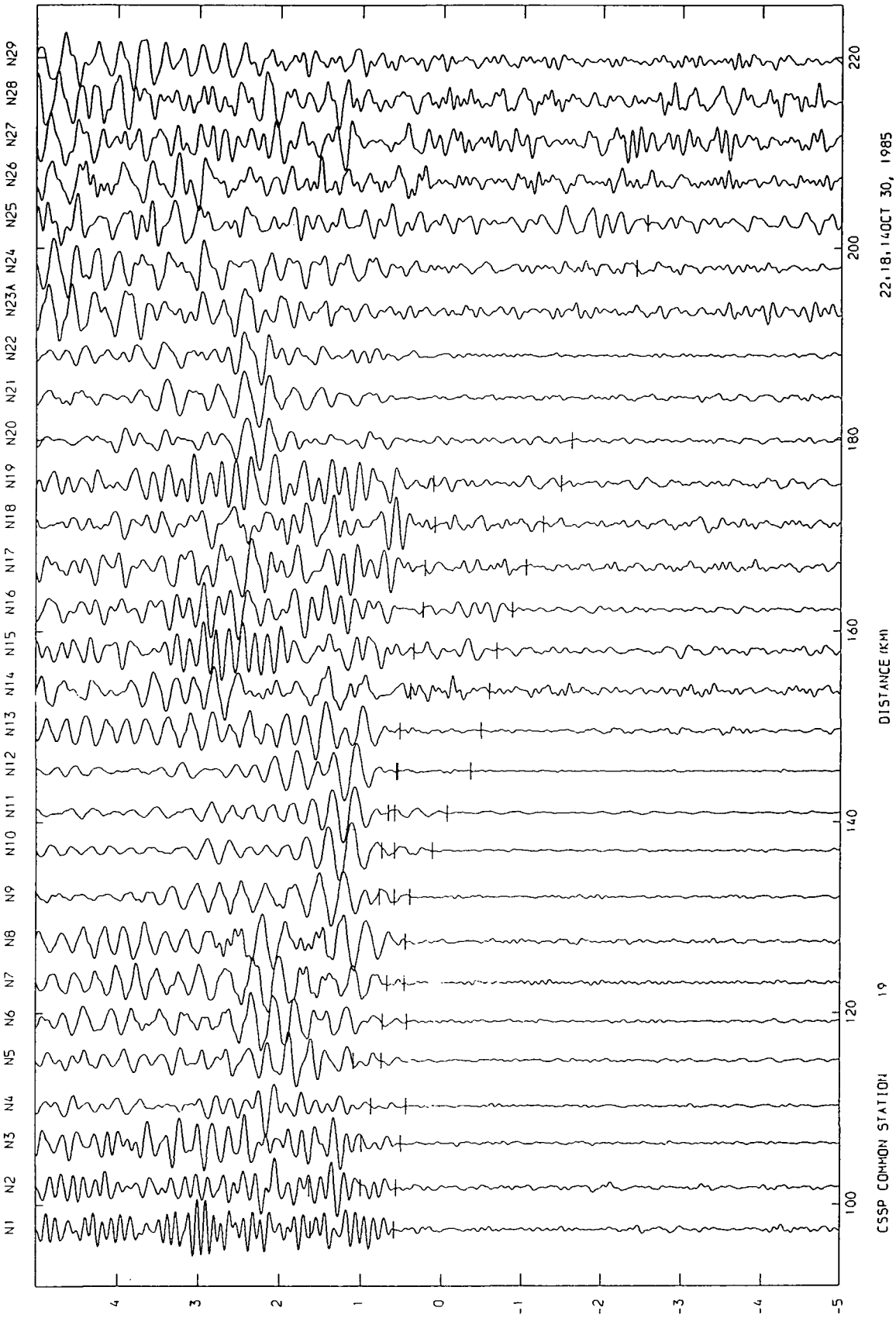
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



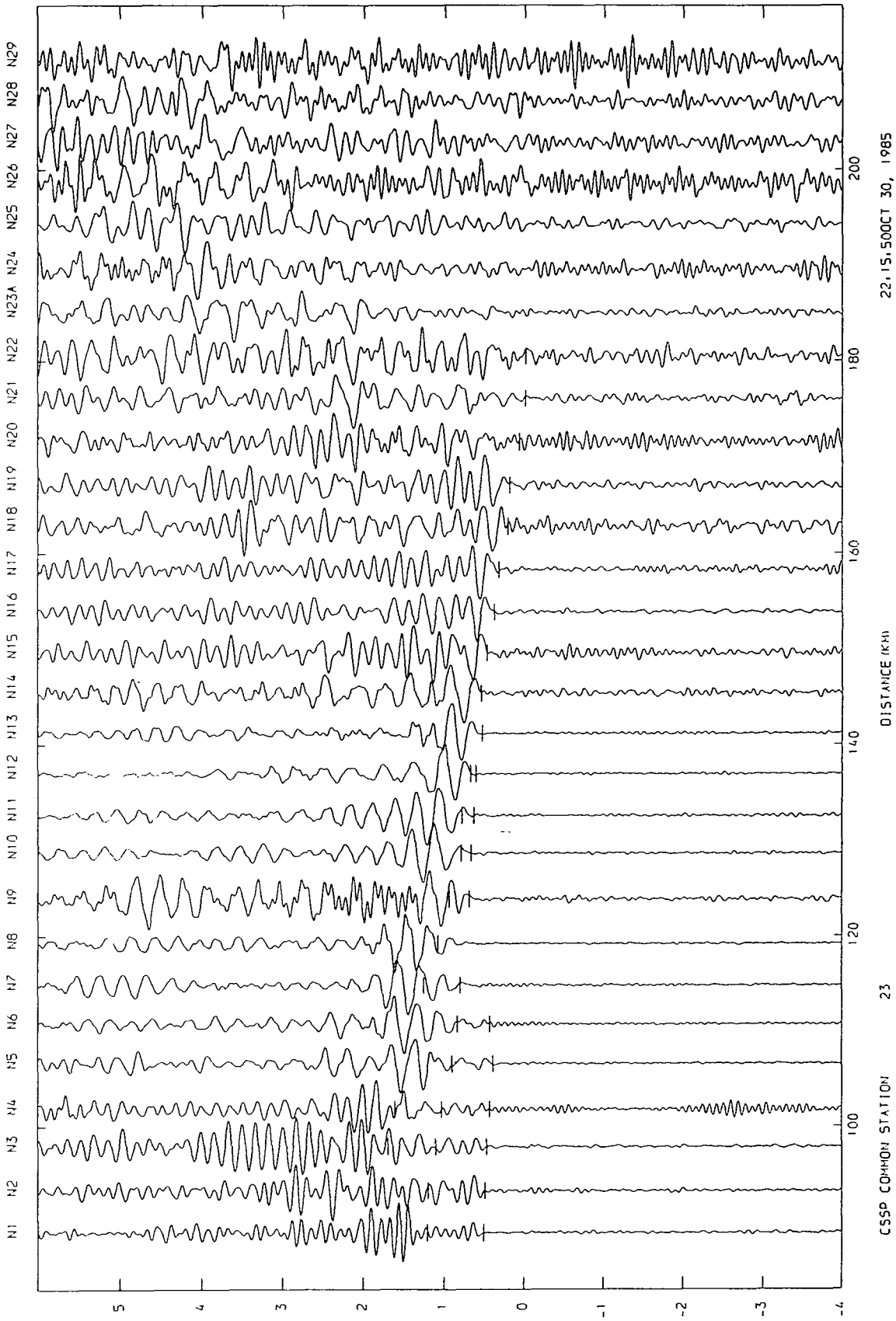
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



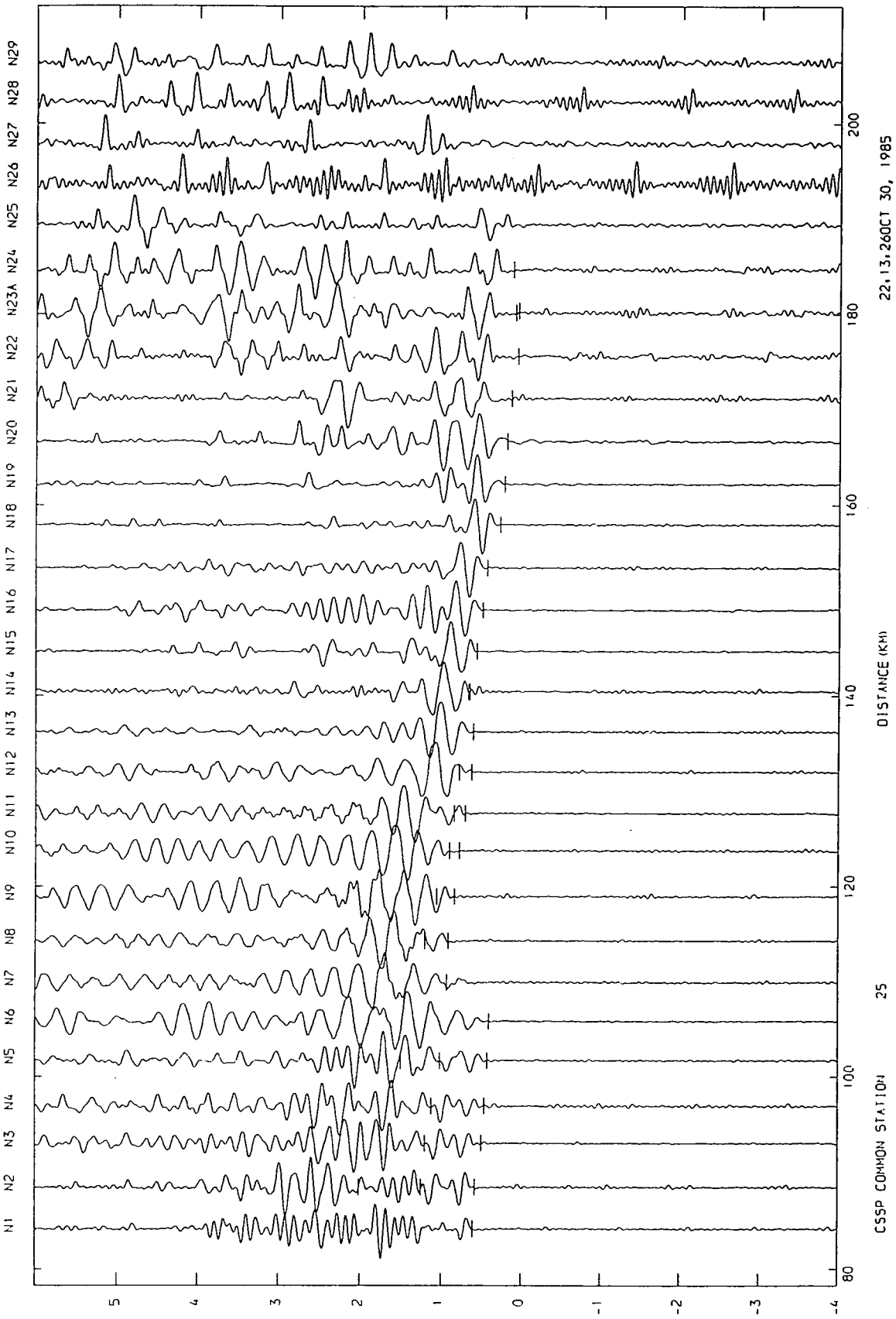
CSSP COMMON STATION 19

22,18,14OCT 30, 1985

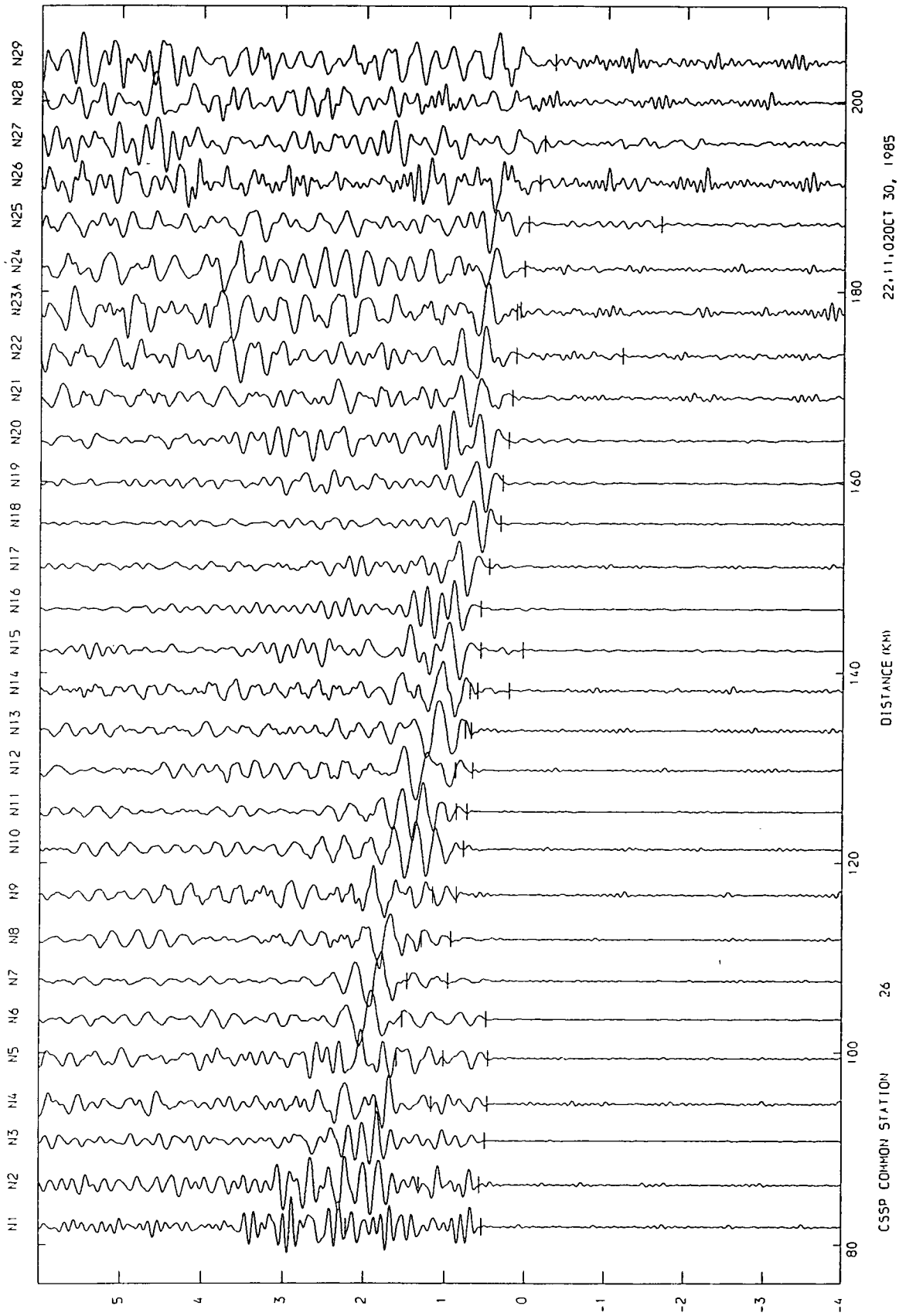
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



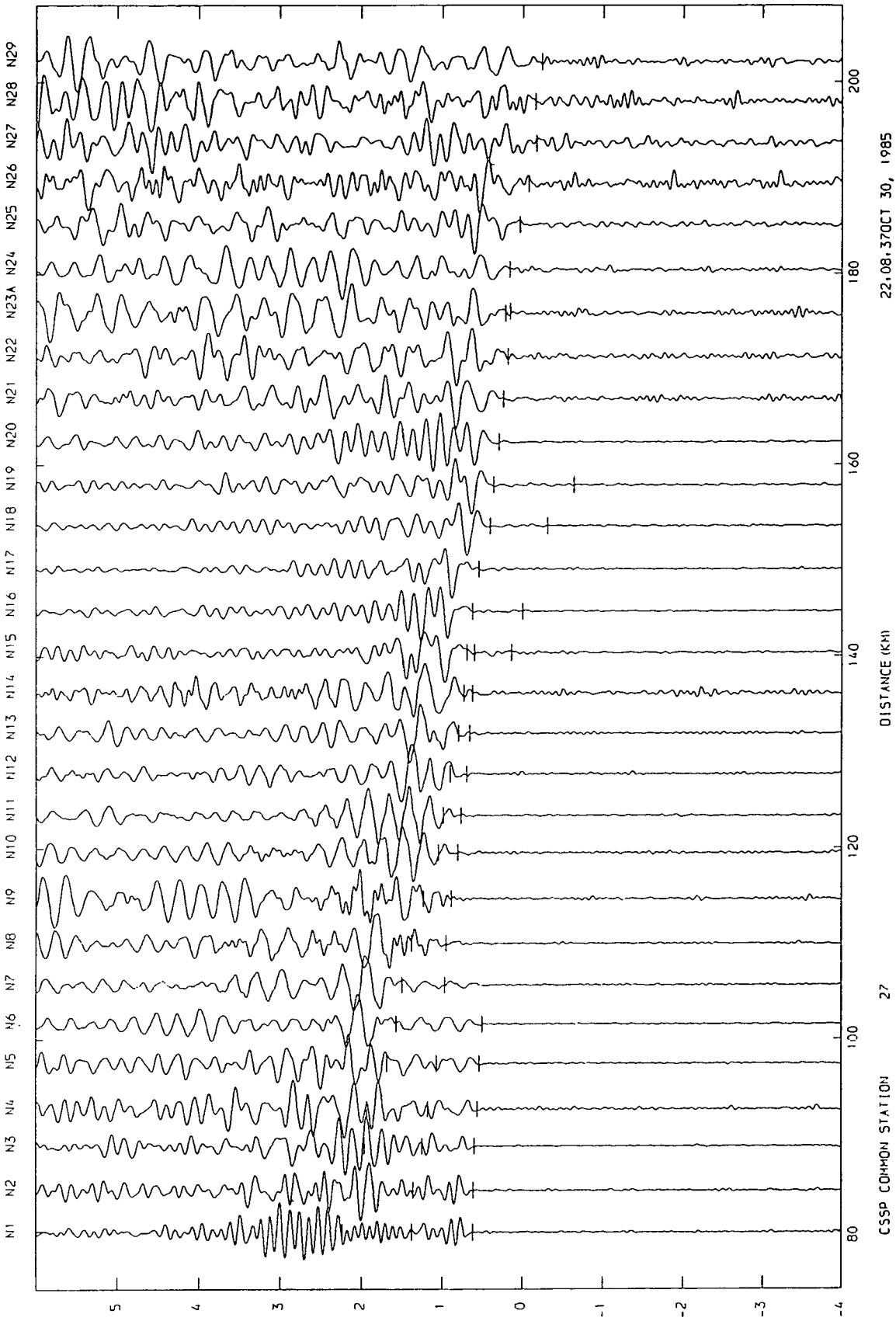
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



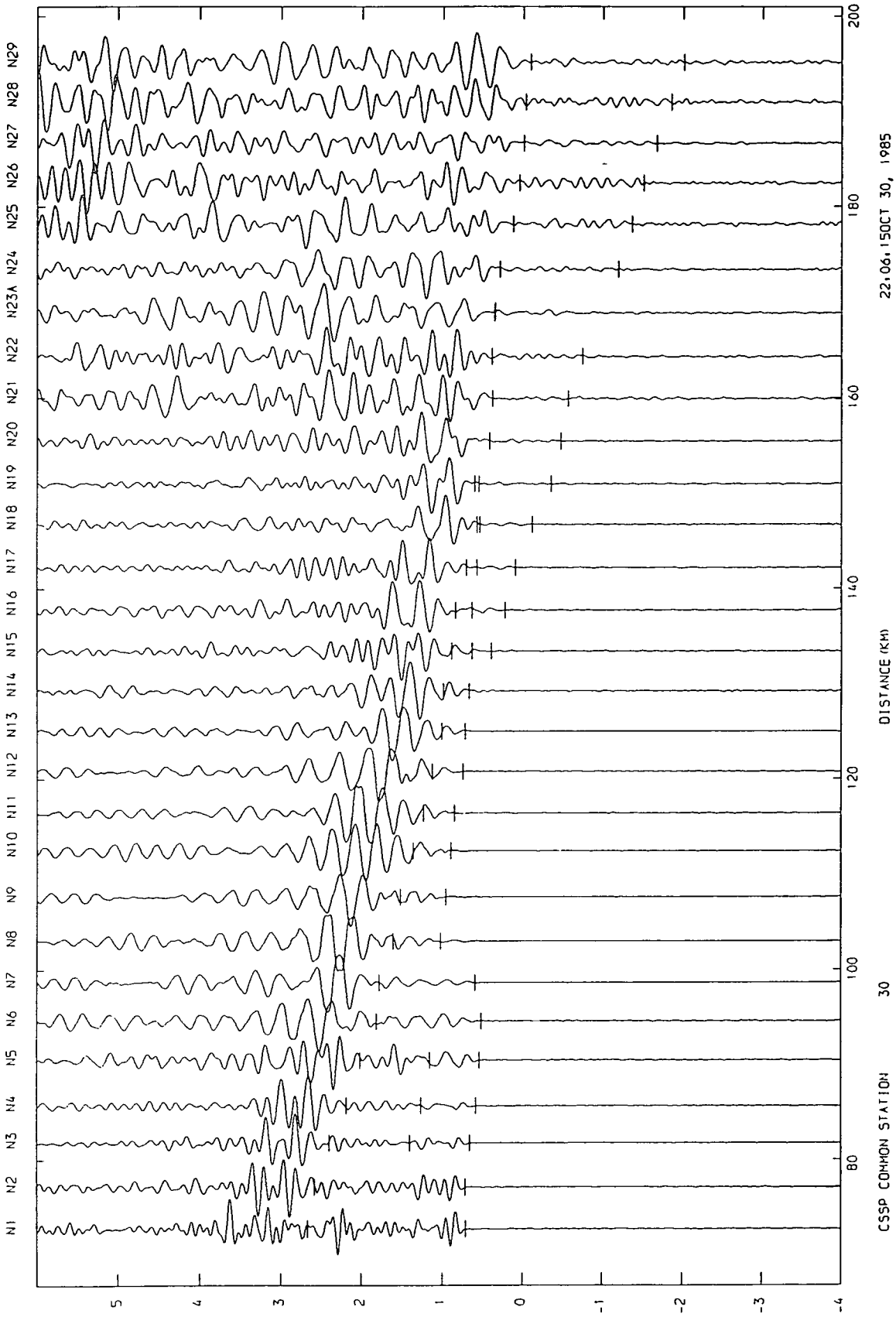
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



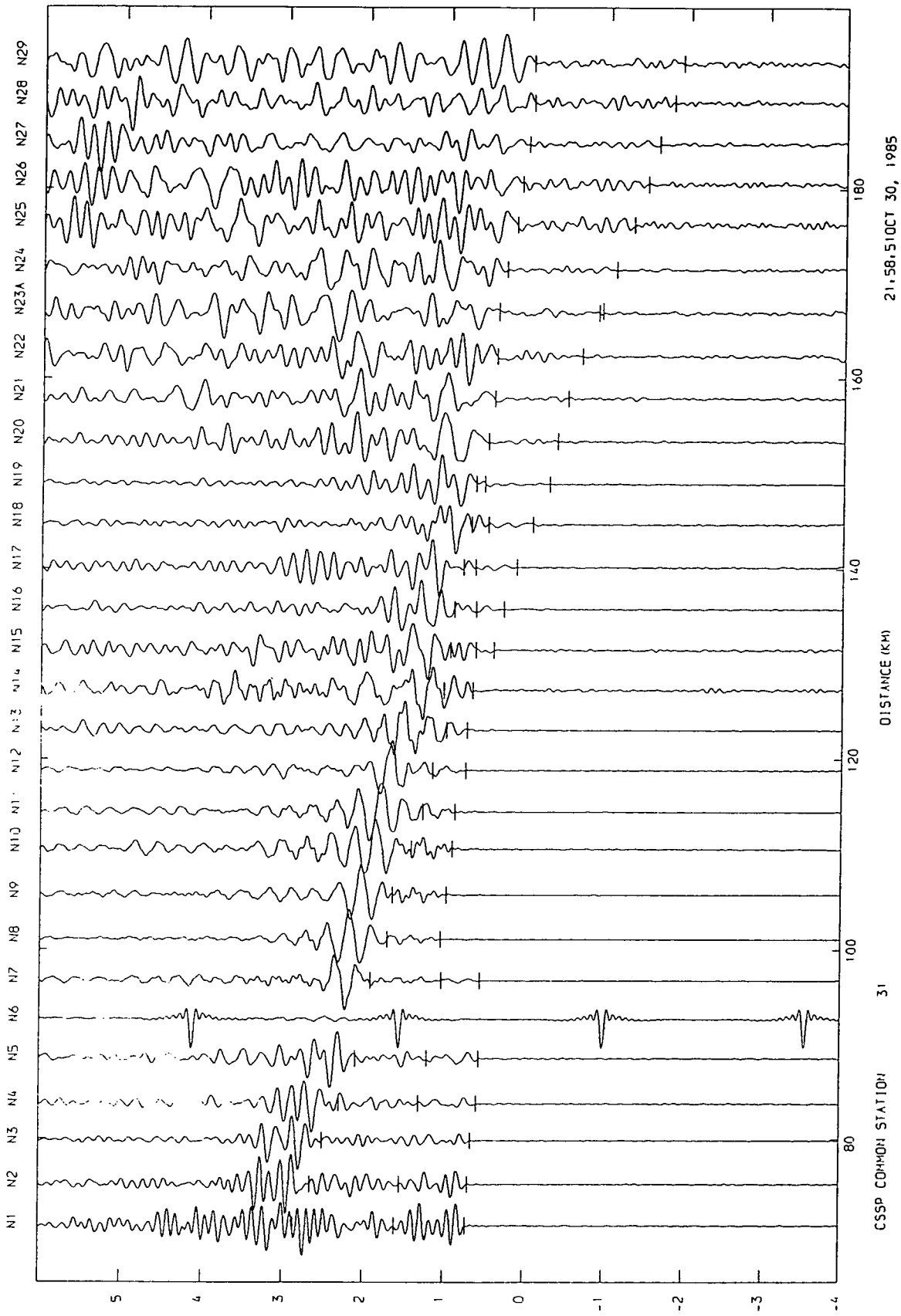
22:08:37OCT 30, 1985

CSSP COMMON STATION 27

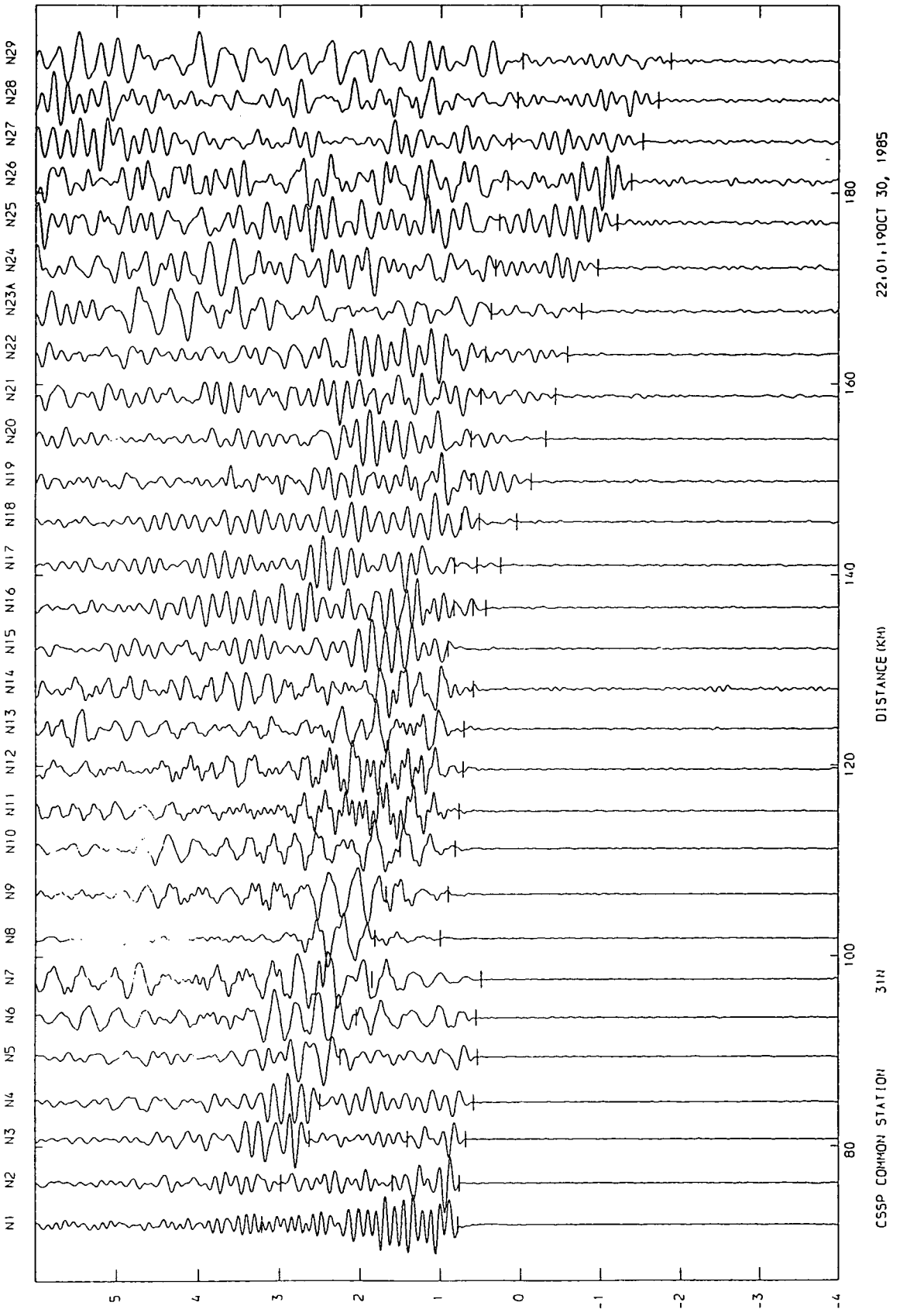
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



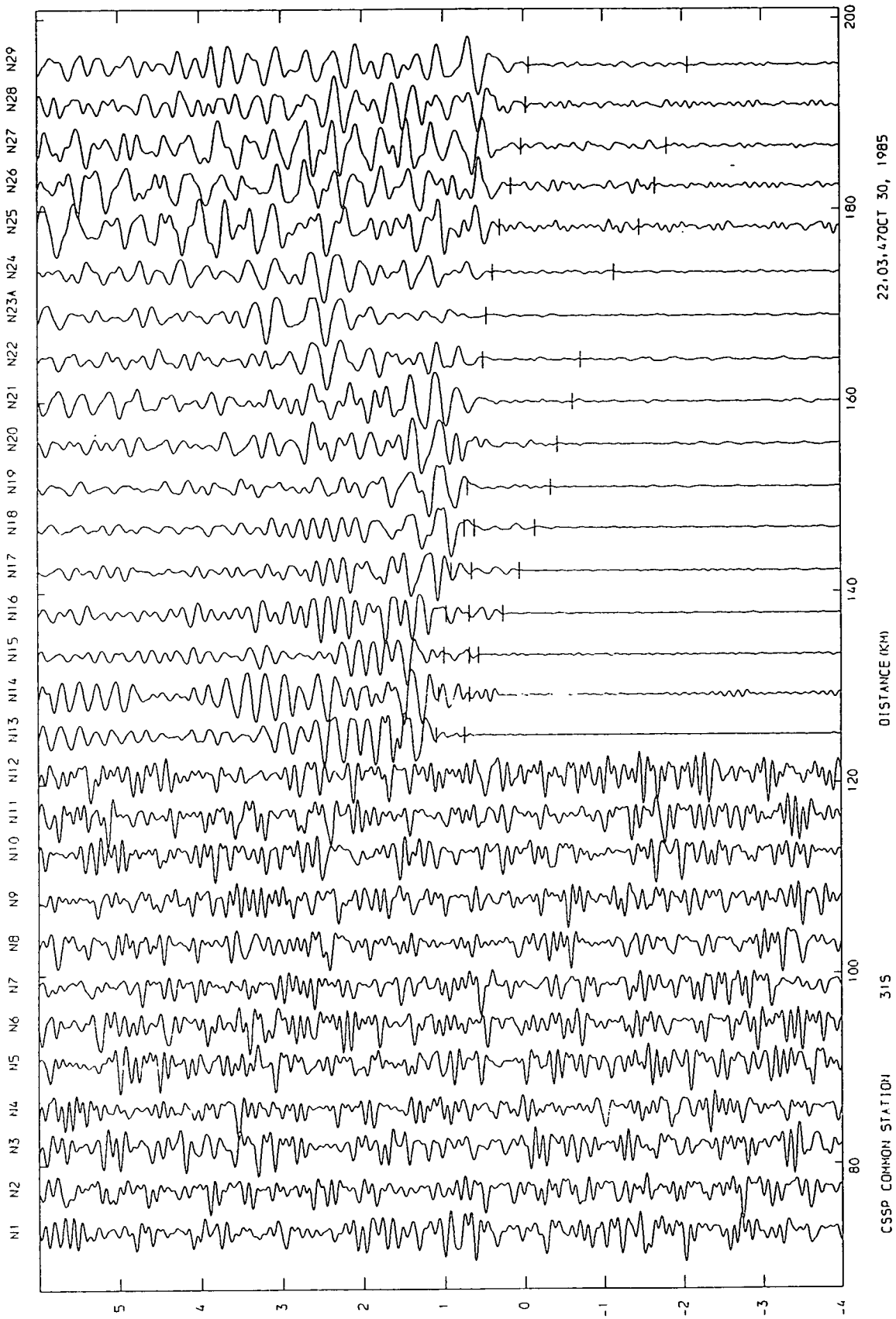
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



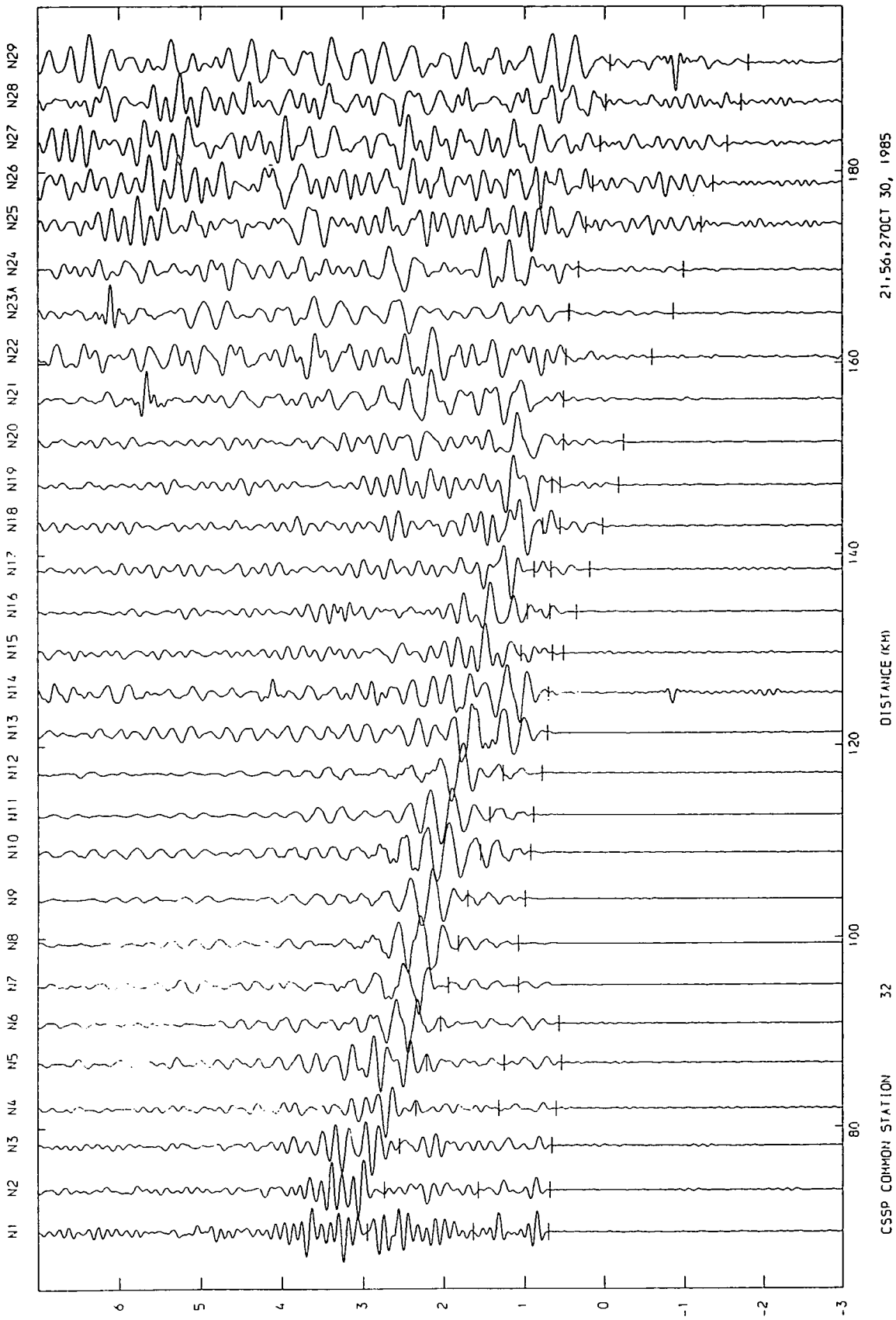
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



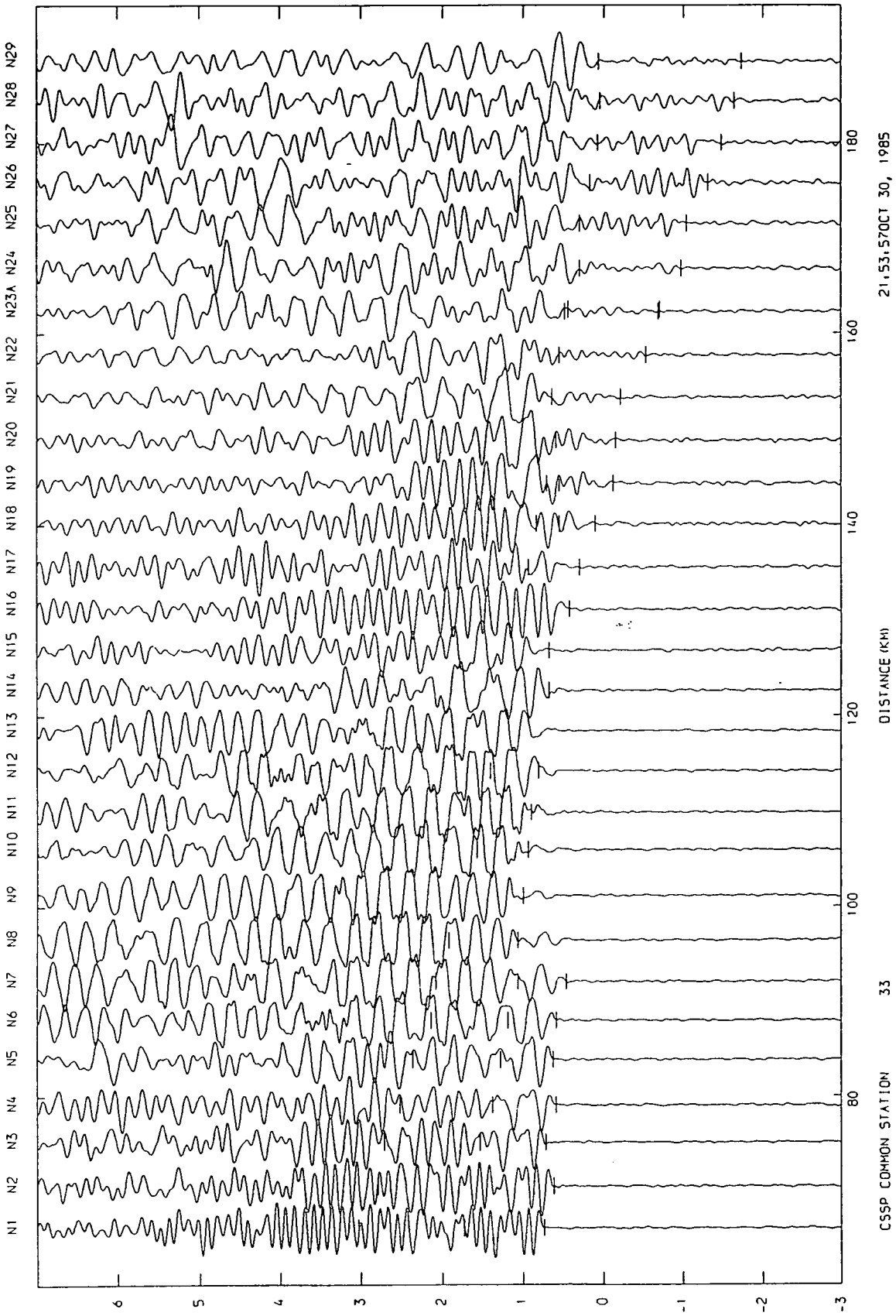
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



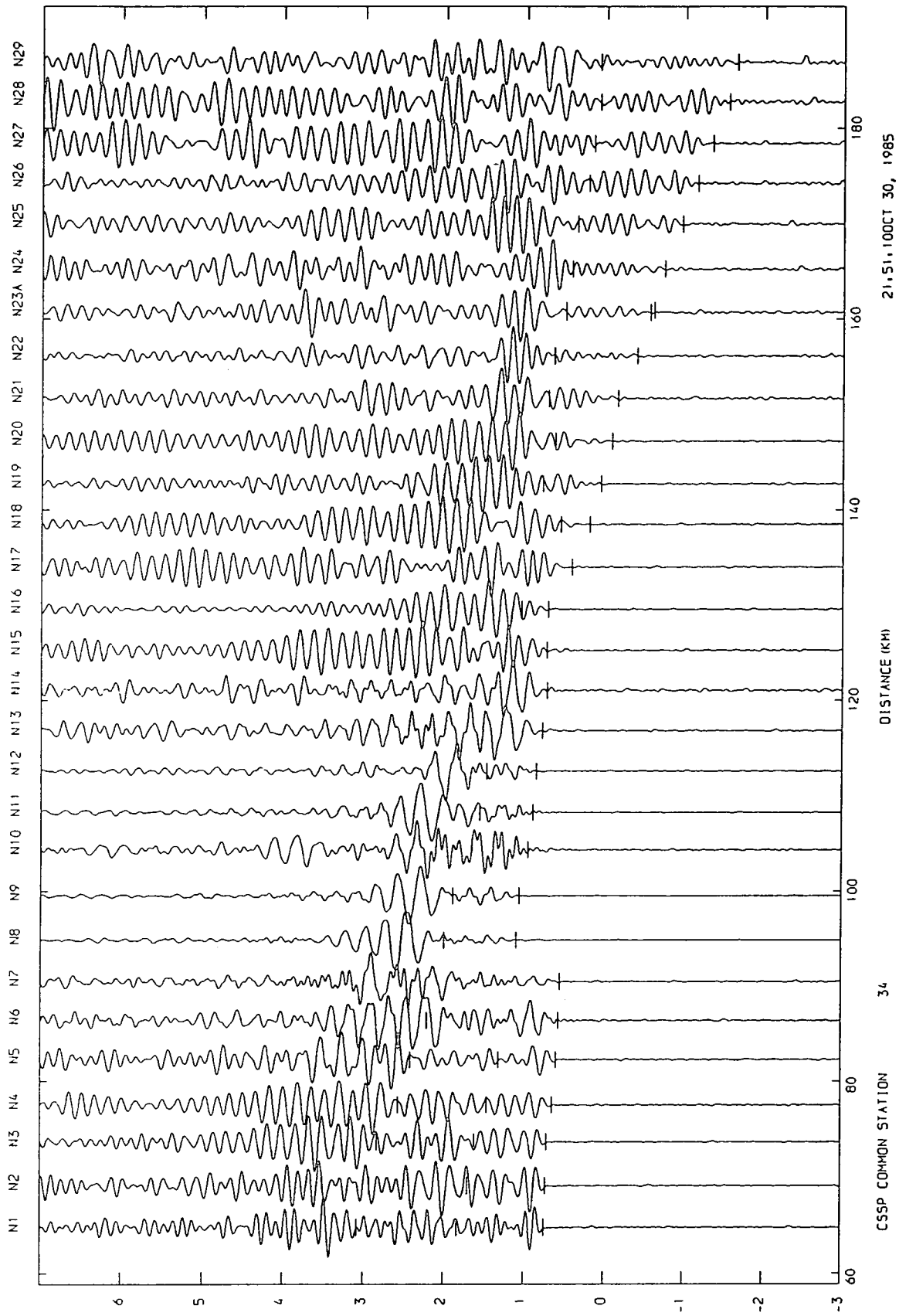
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



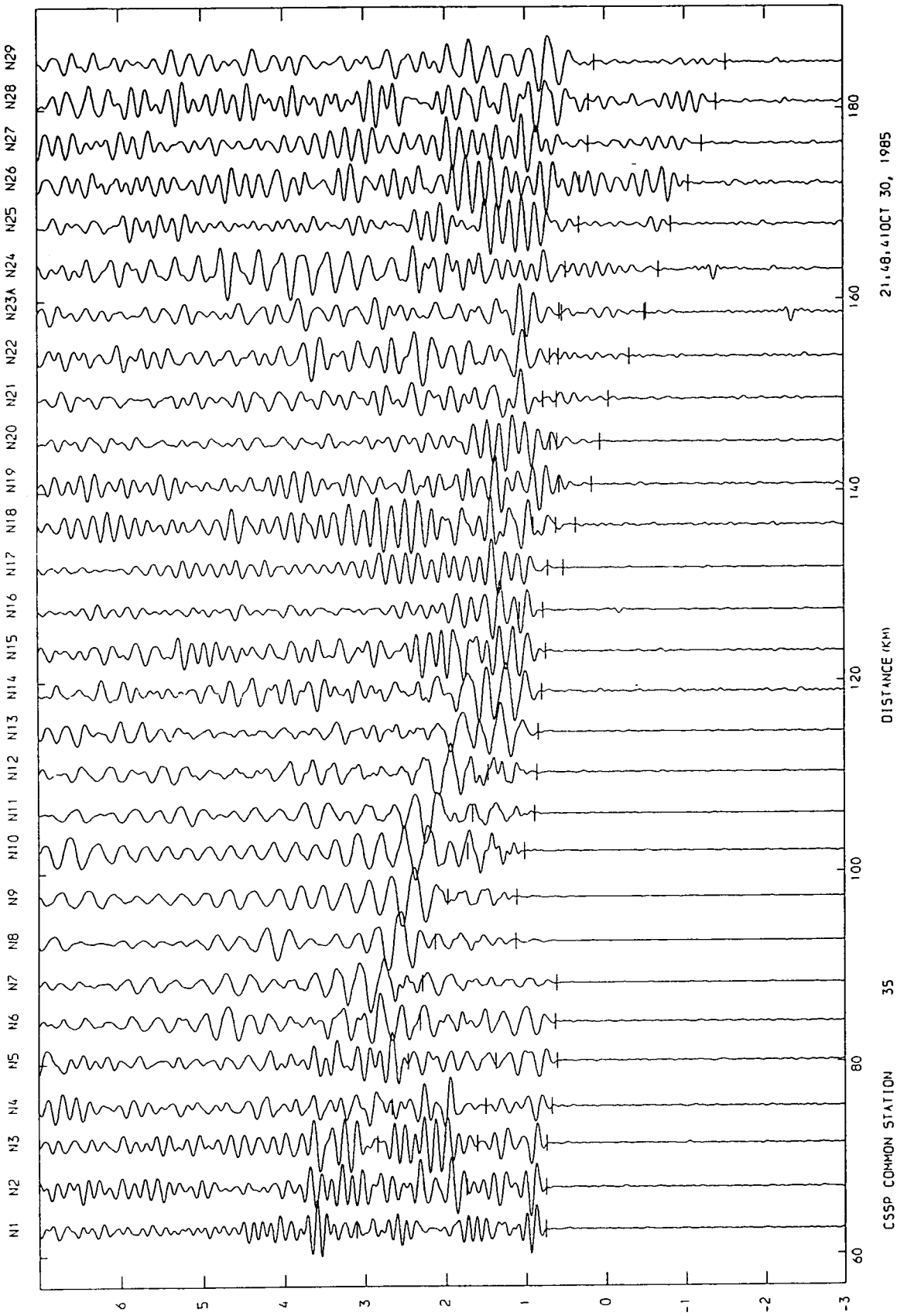
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



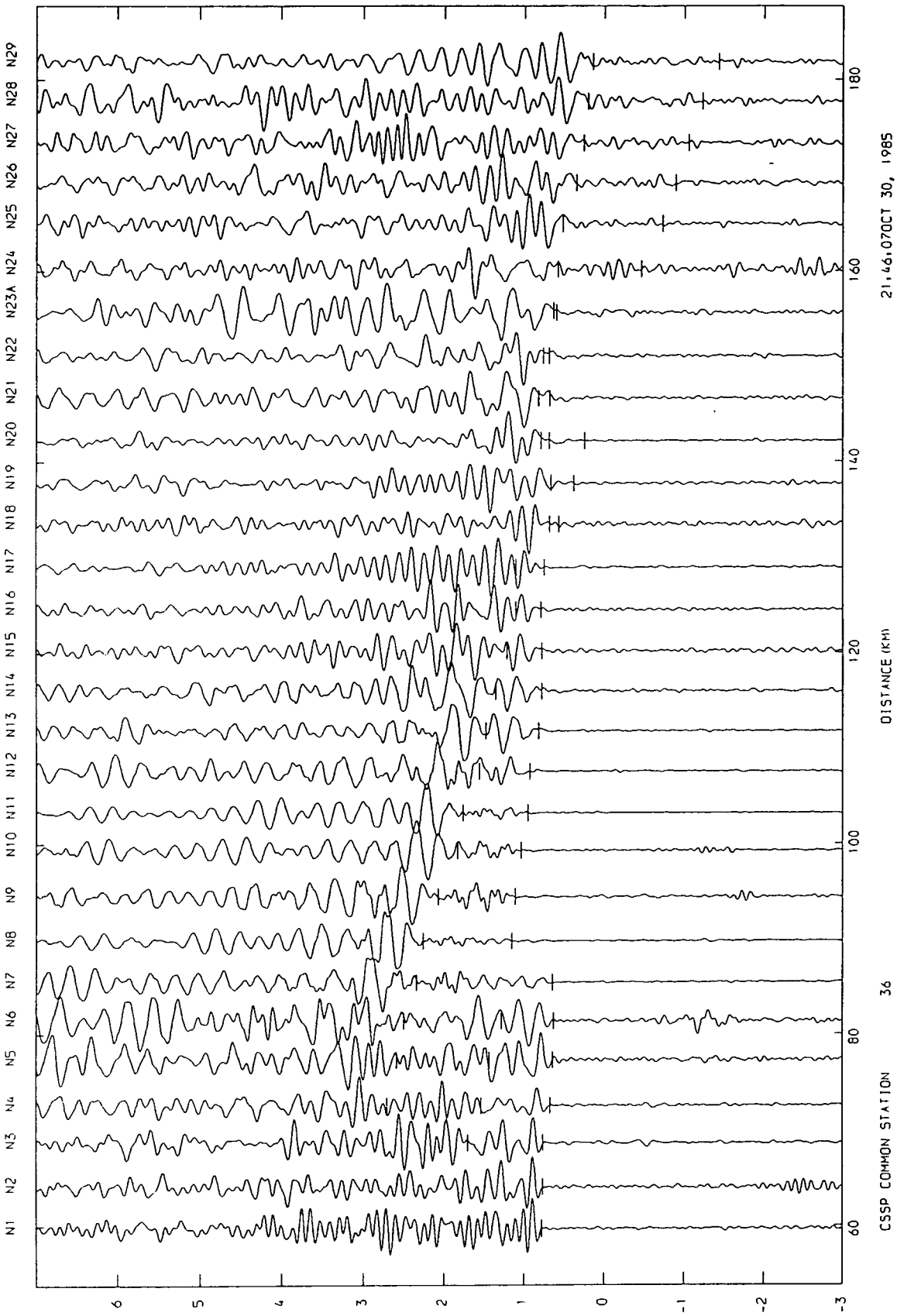
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



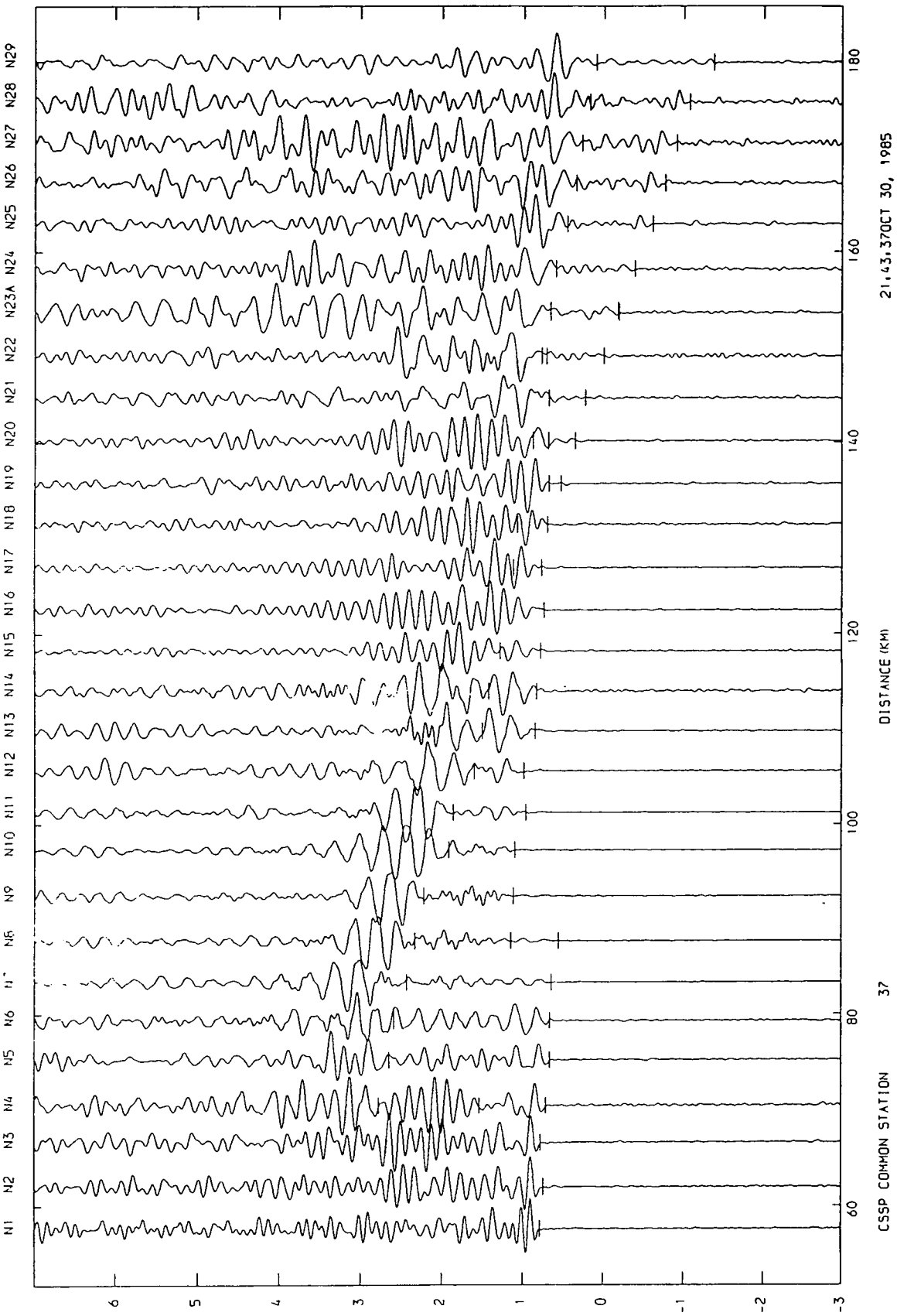
CSSP COMMON STATION 35

21:48:41OCT 30, 1985

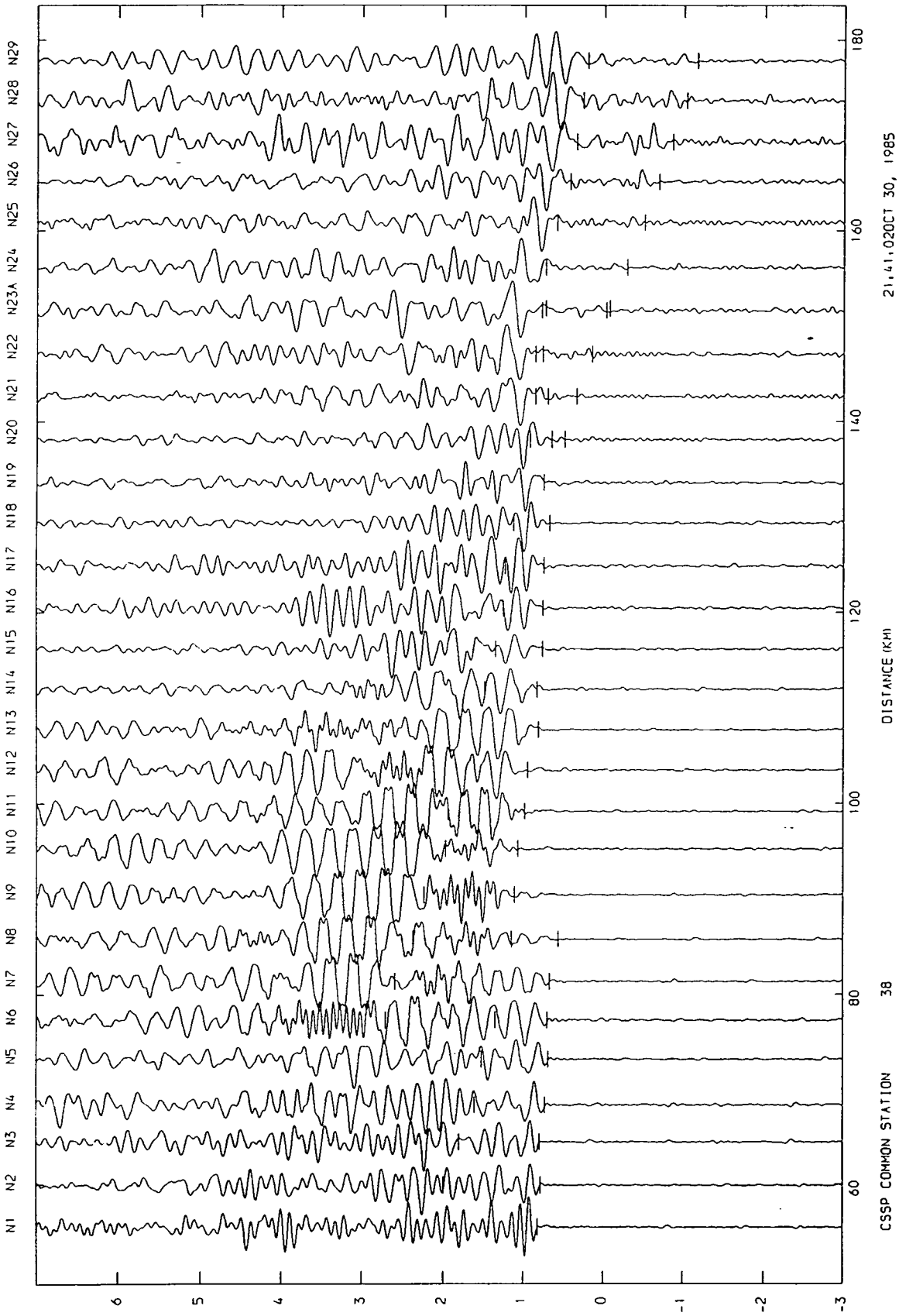
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



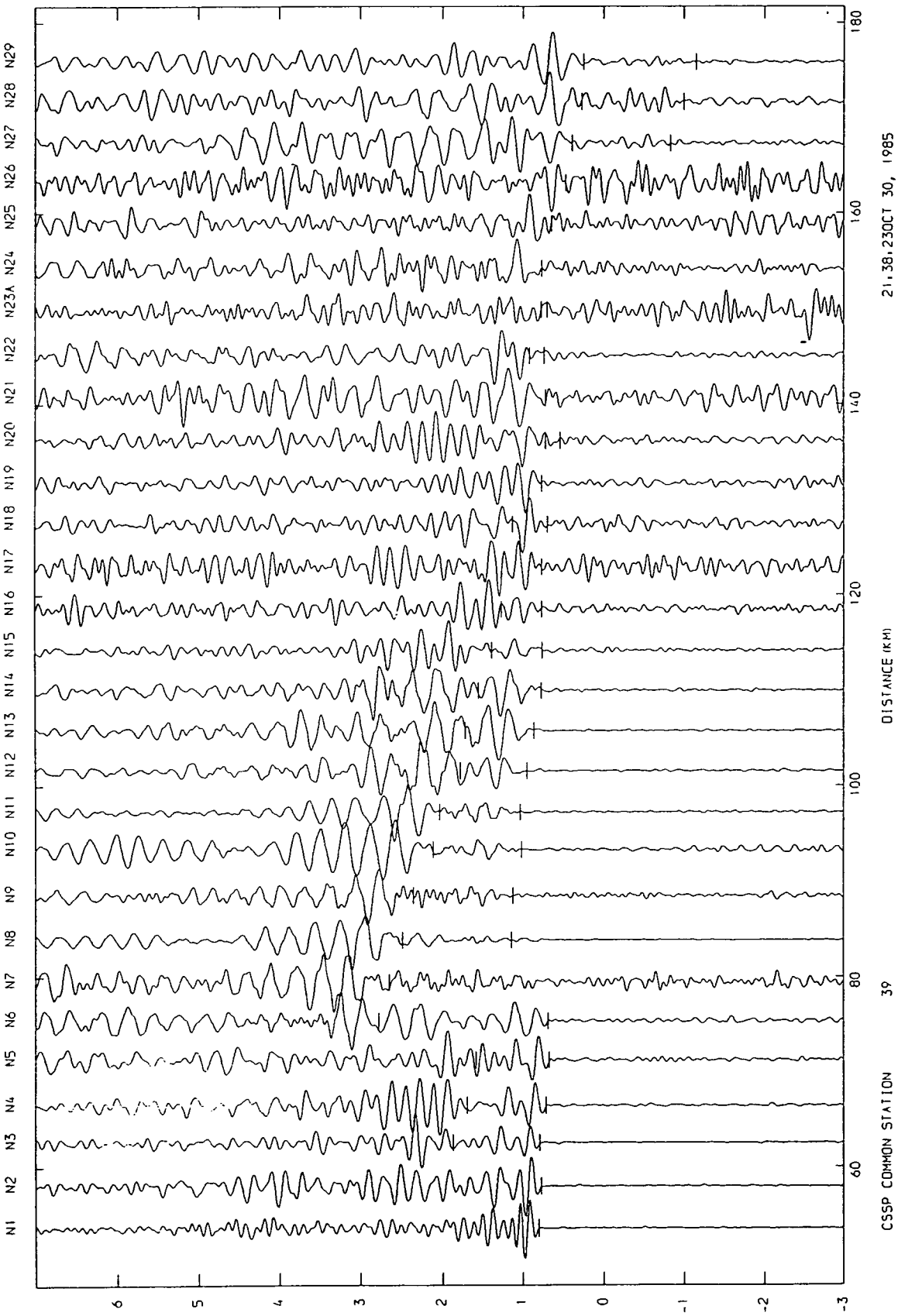
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



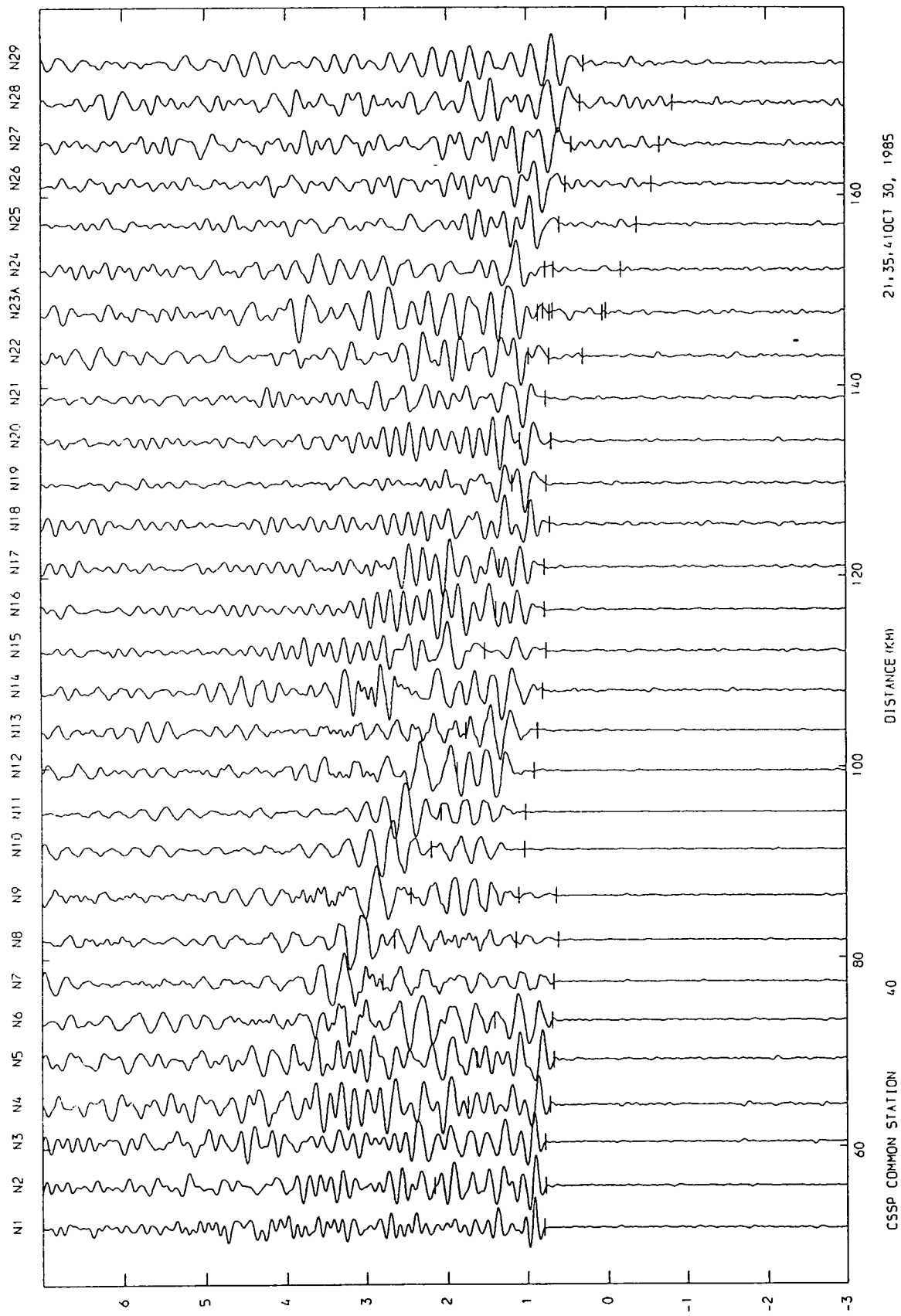
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

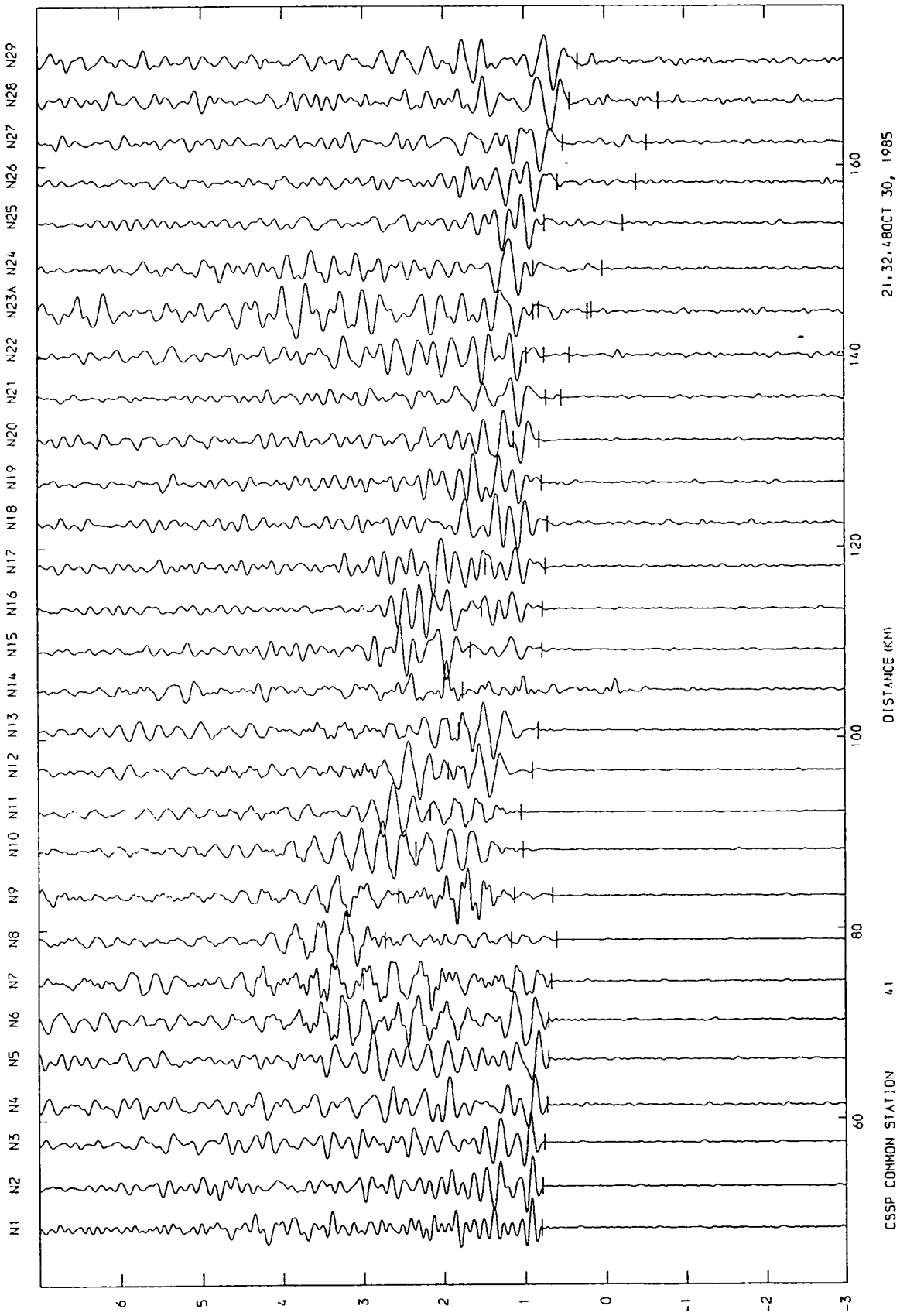


FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

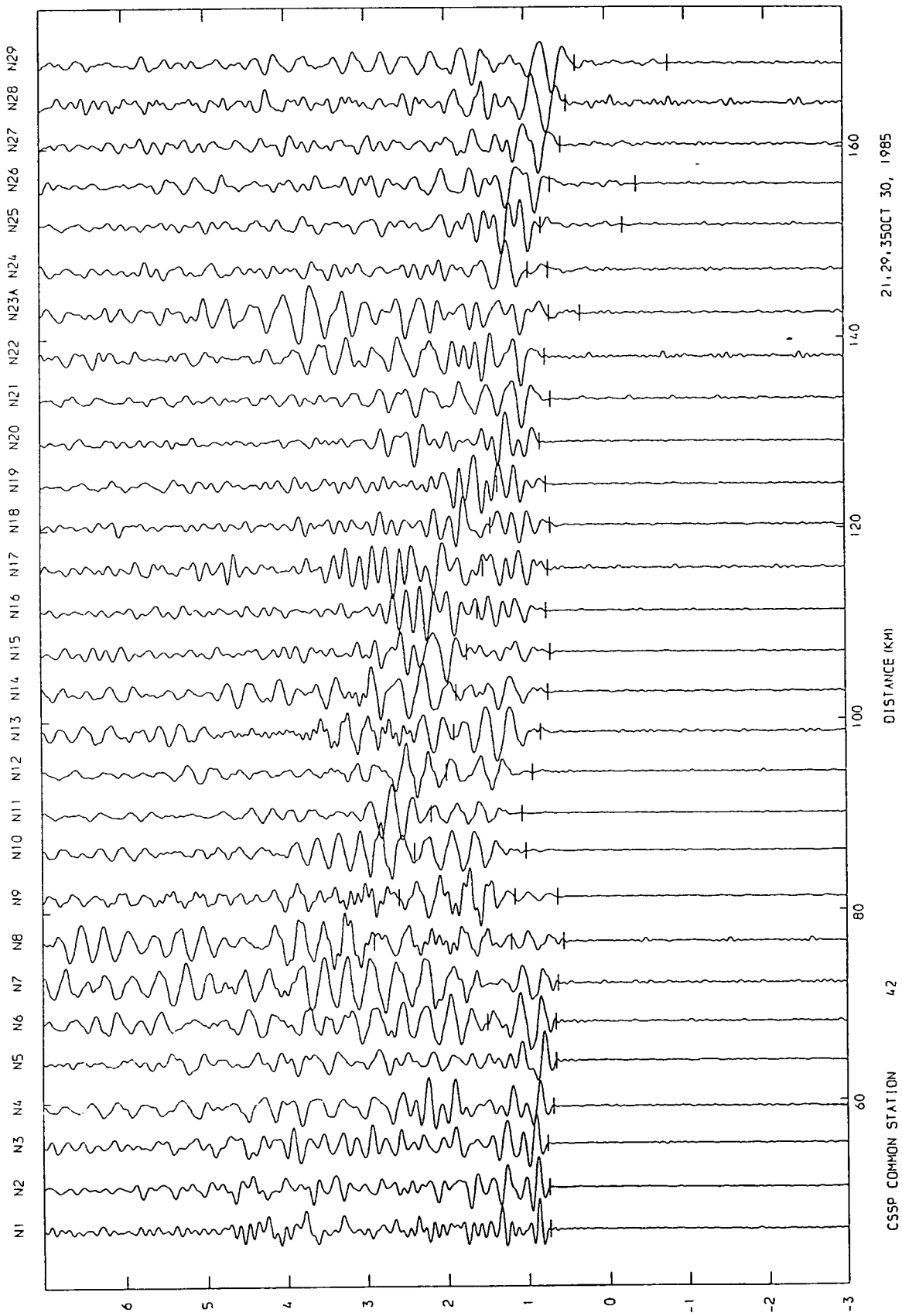


UNIV DUNEDIN

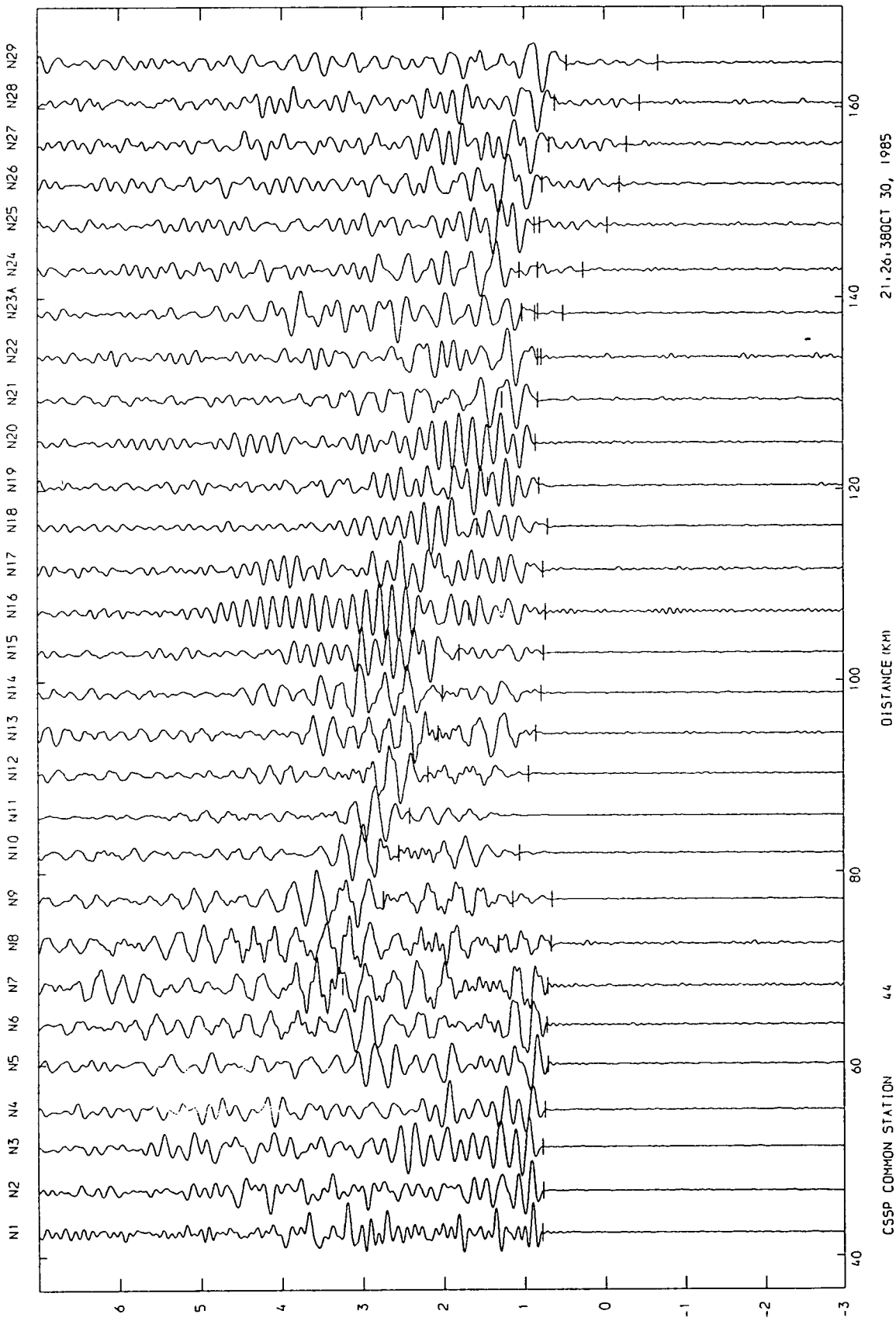
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



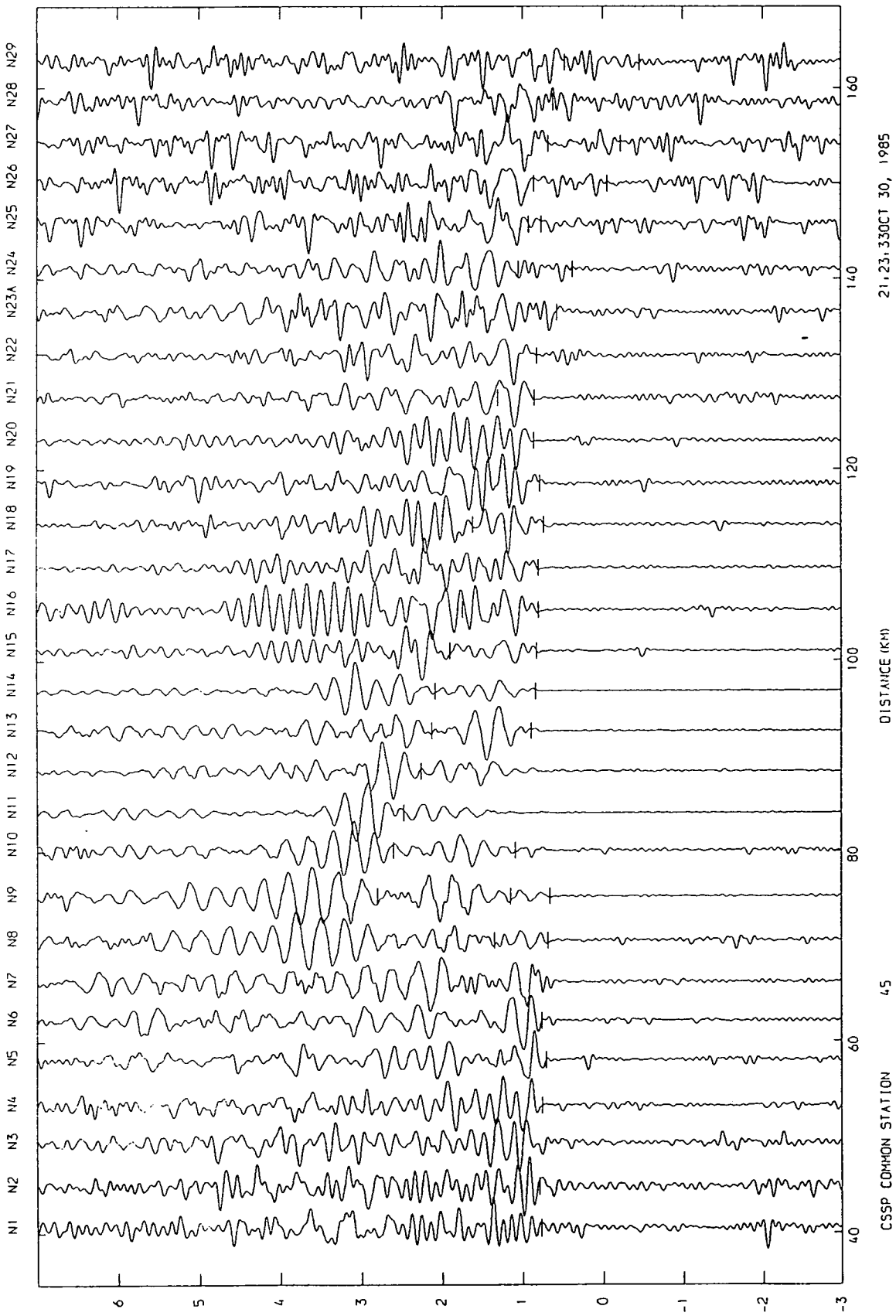
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



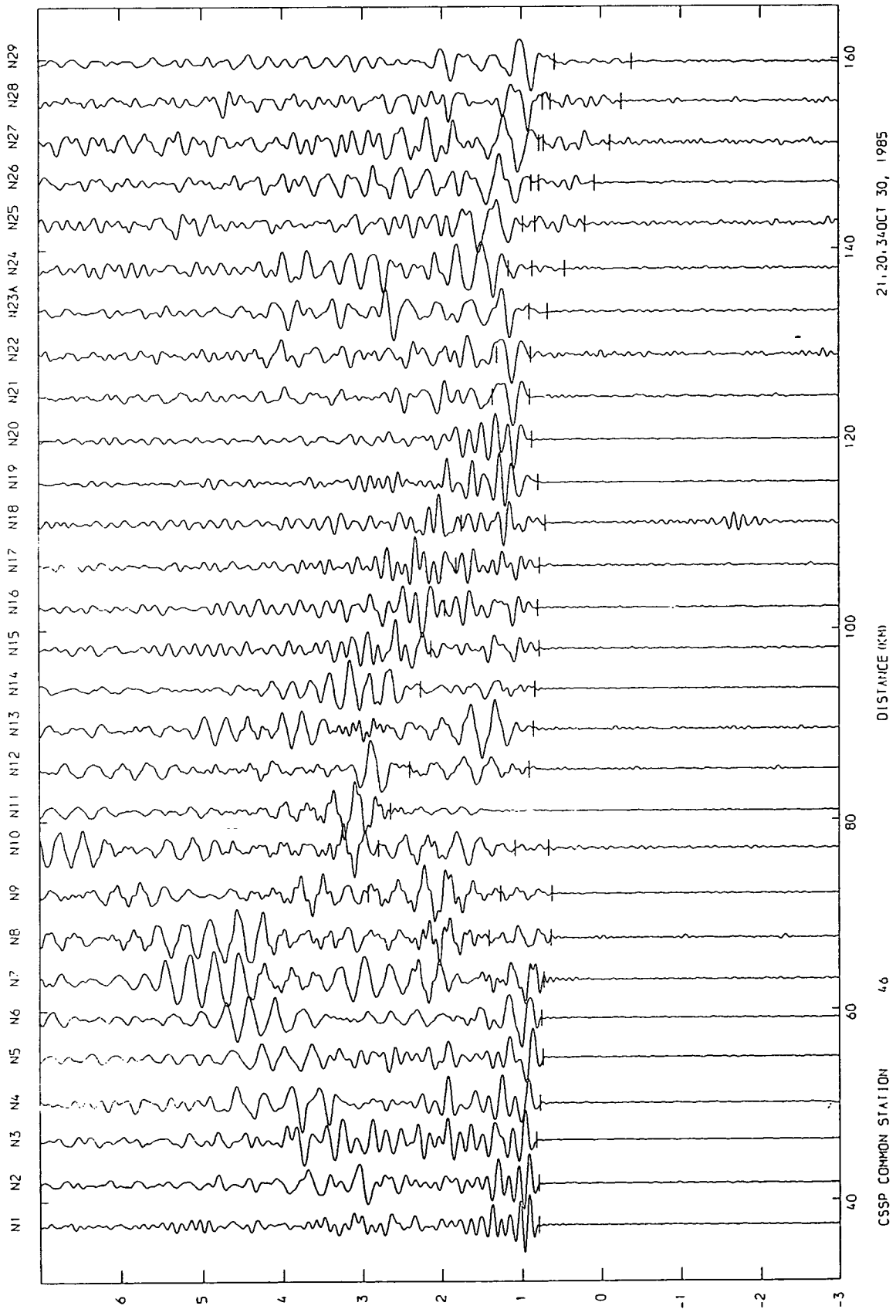
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



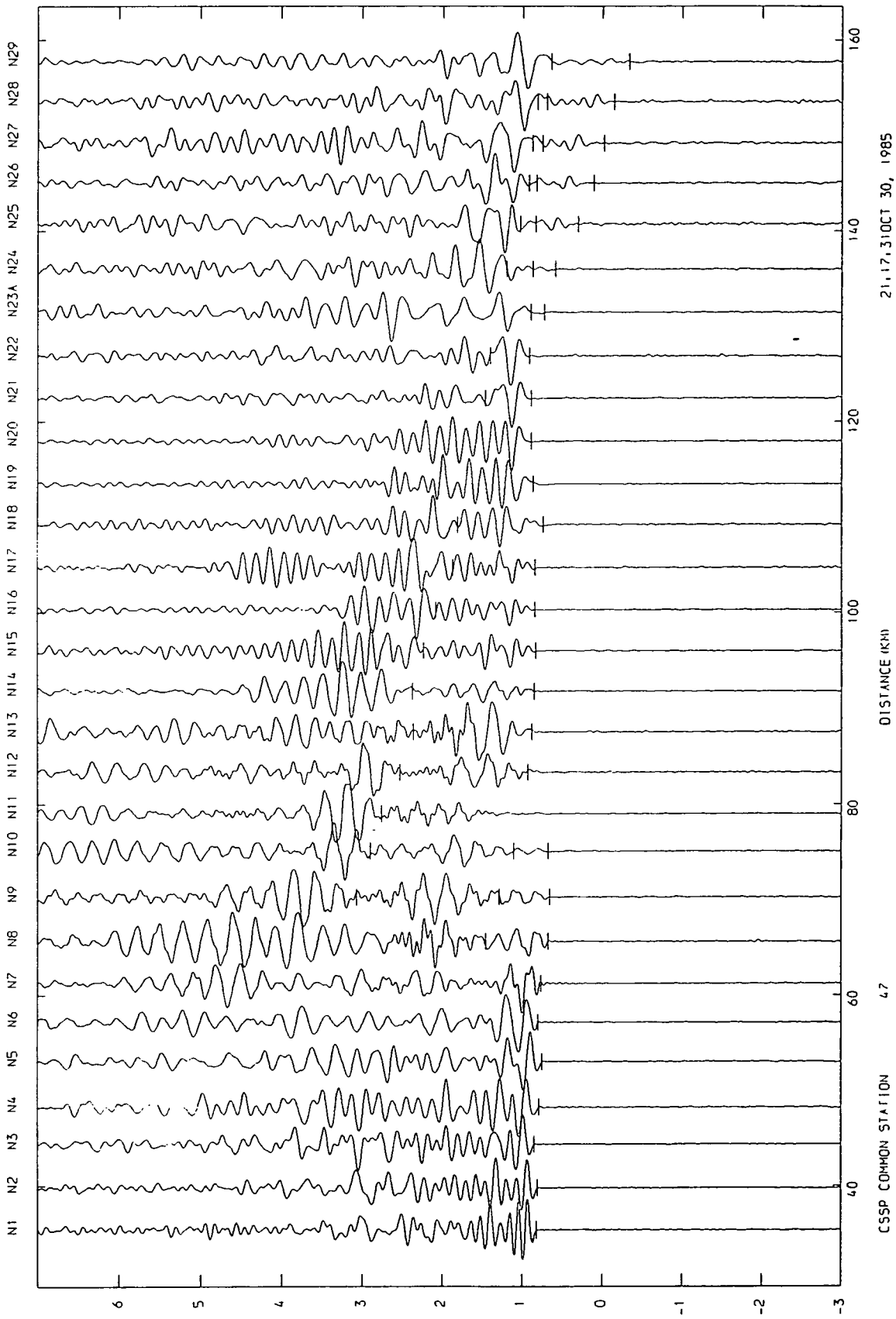
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



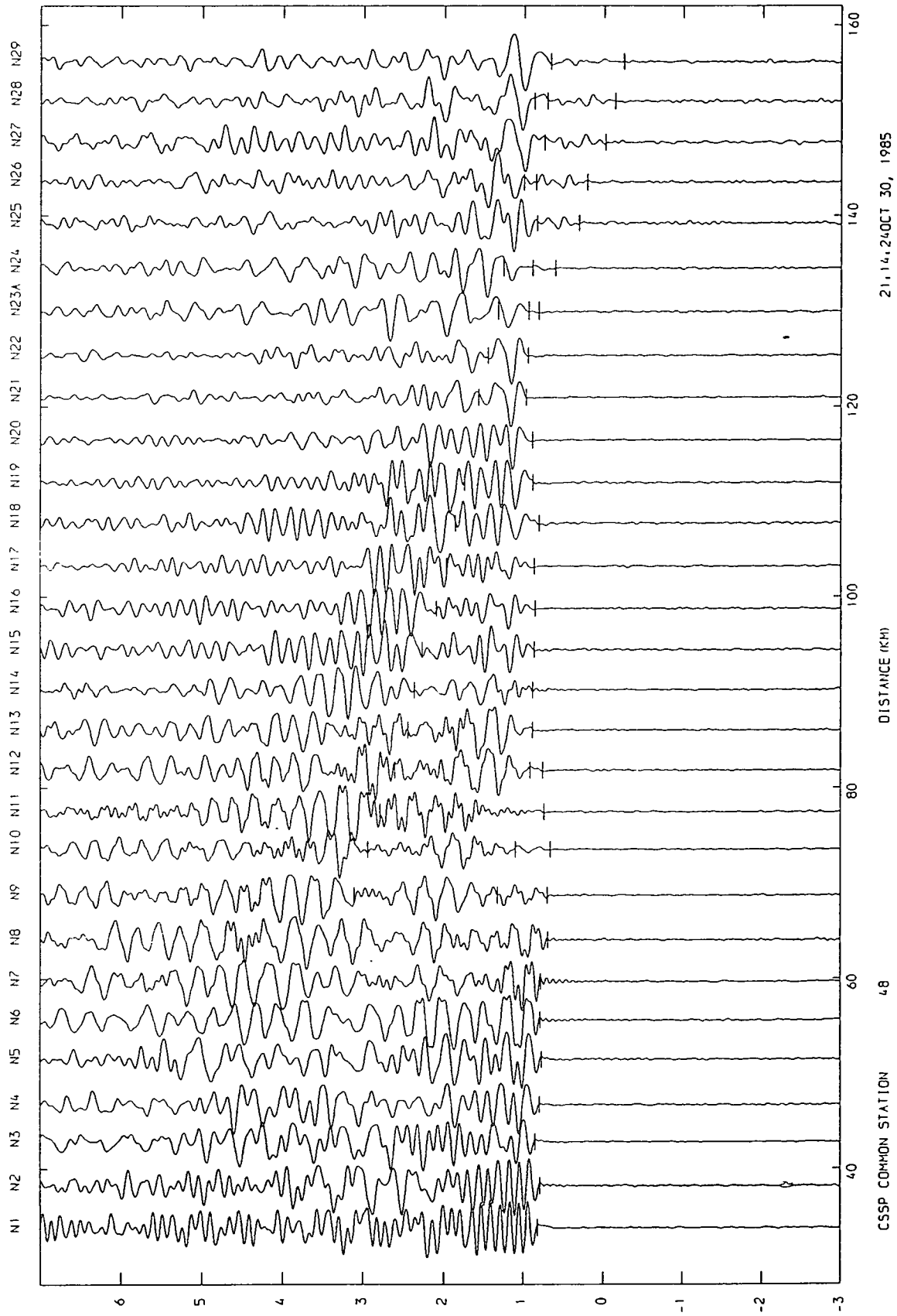
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



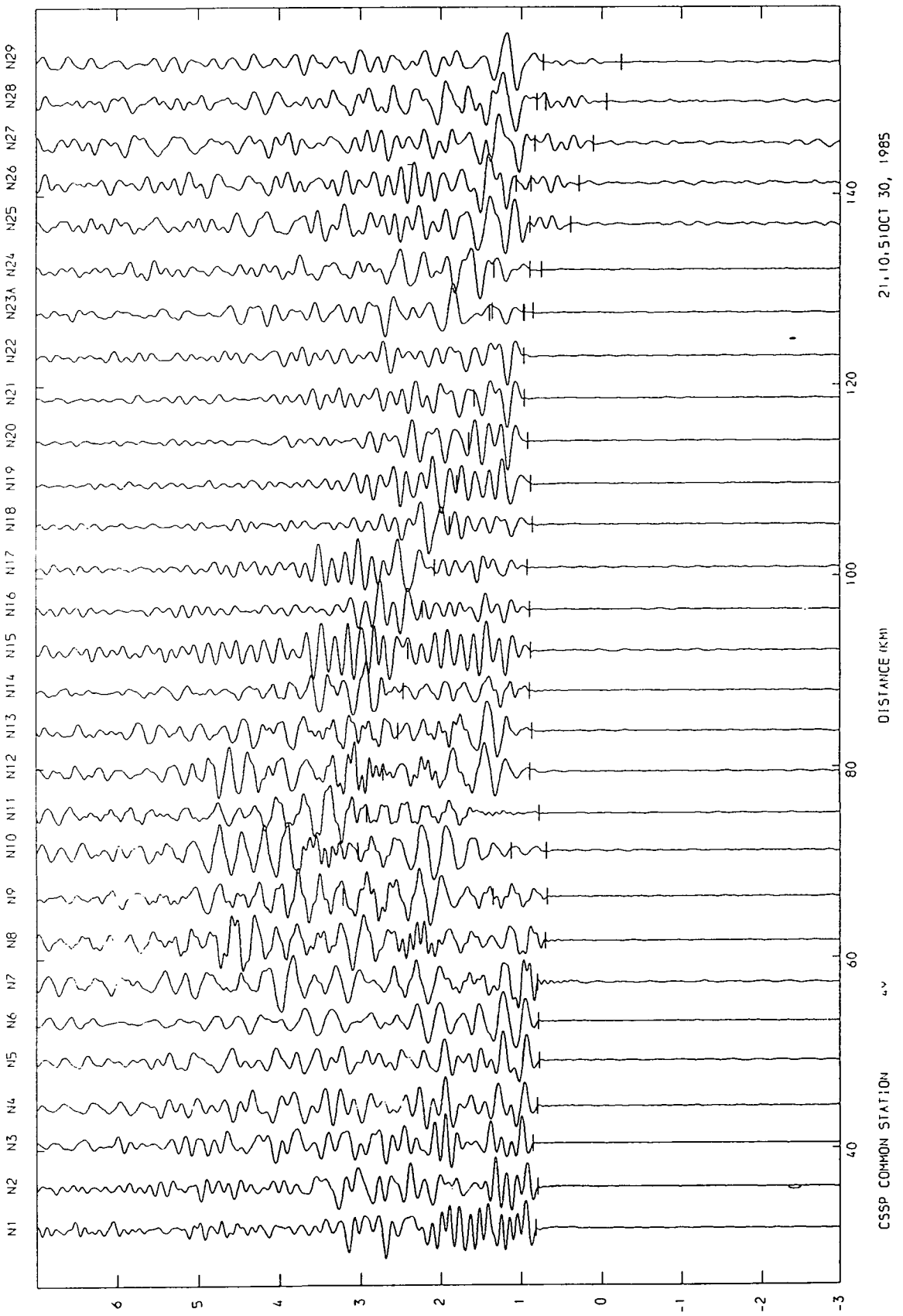
21:17:31OCT 30, 1985

CSSP COMMON STATION 47

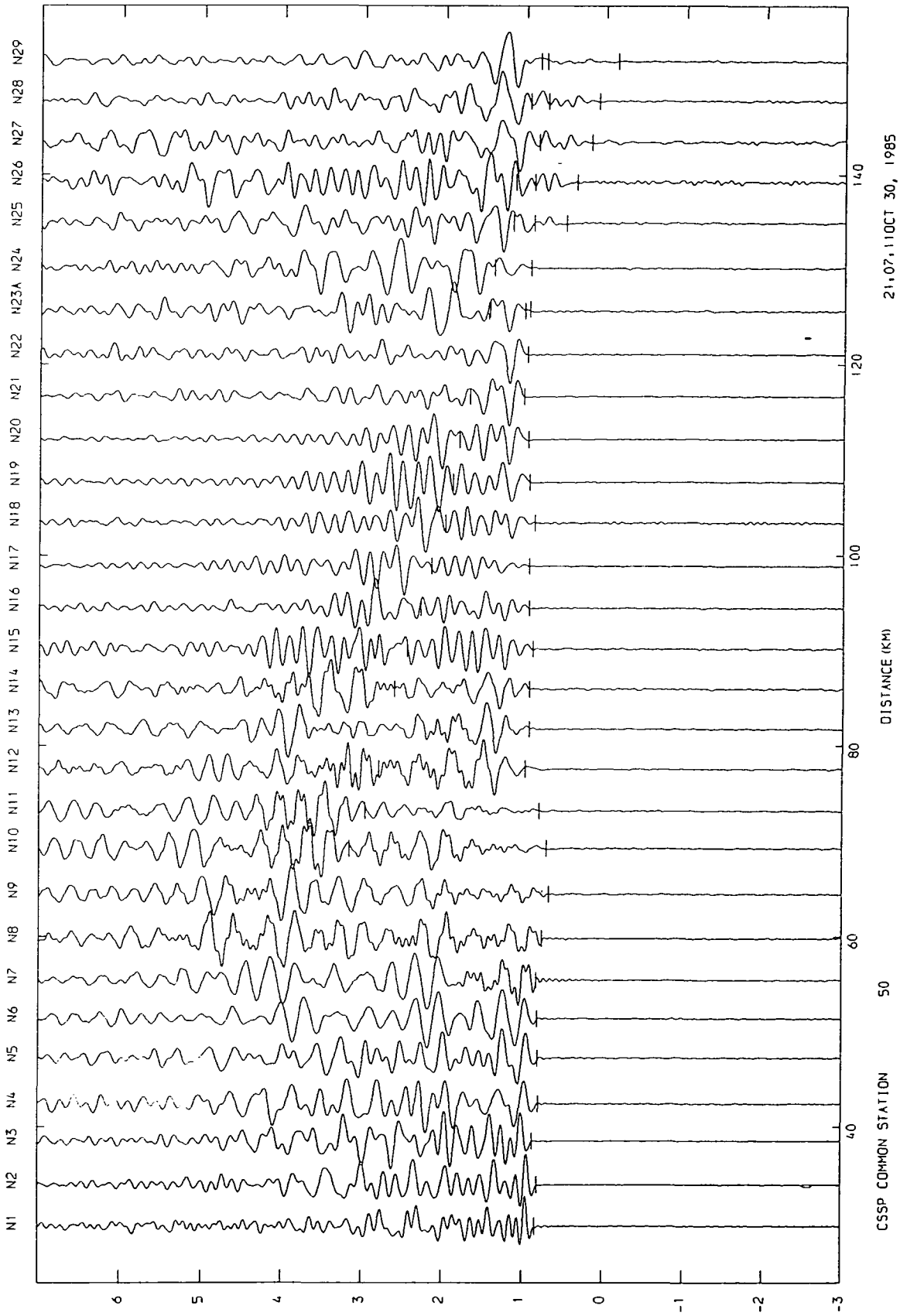
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



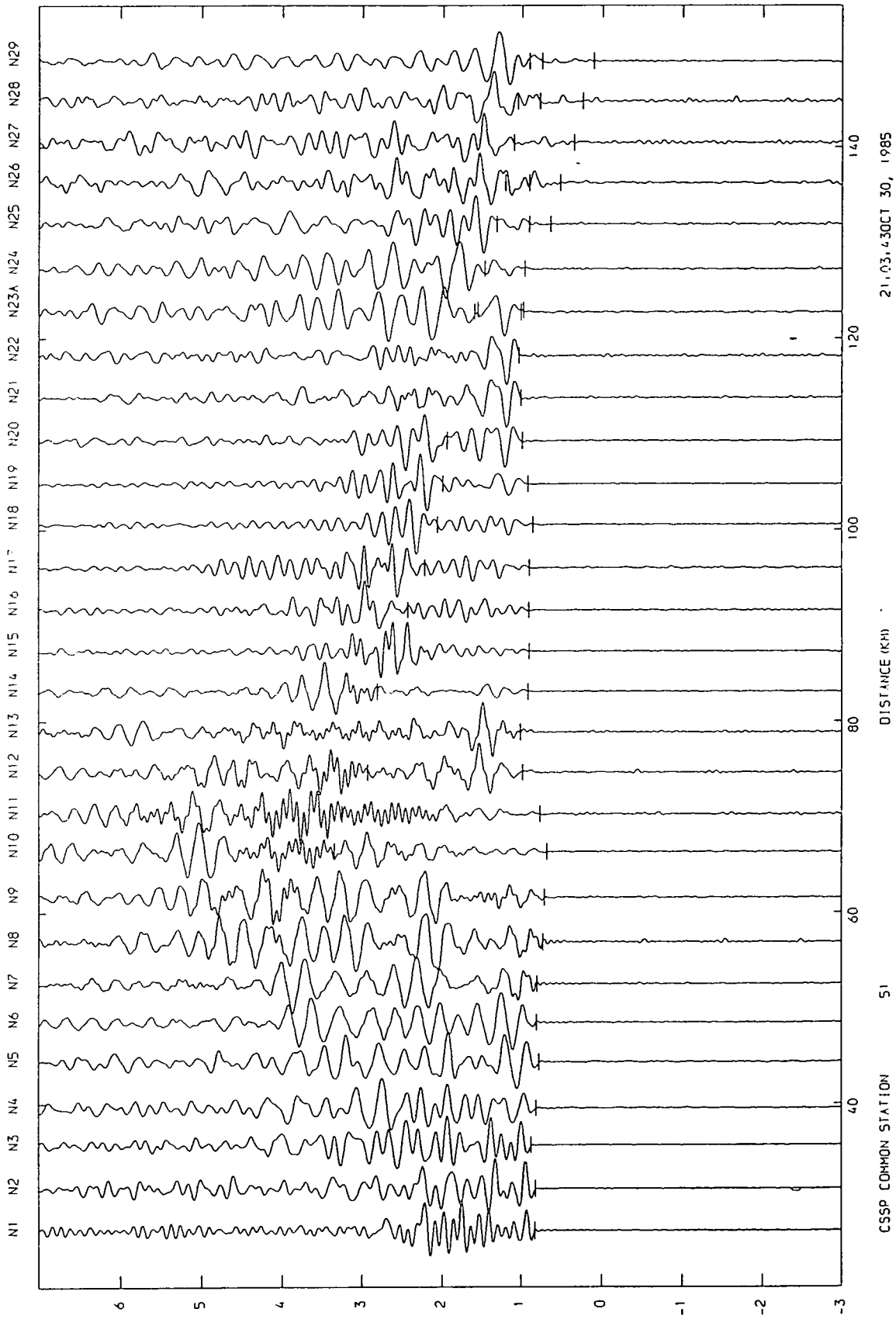
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



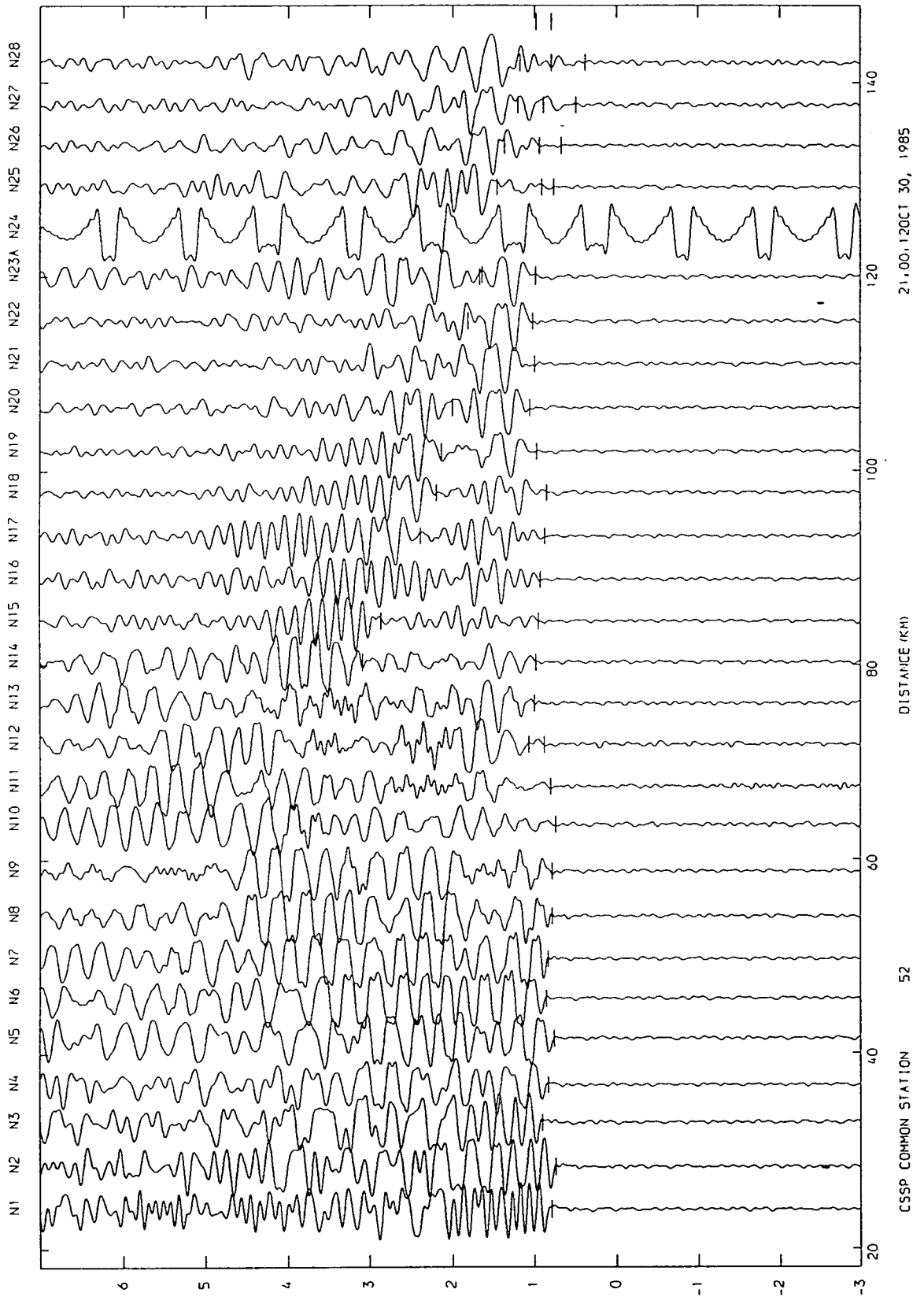
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



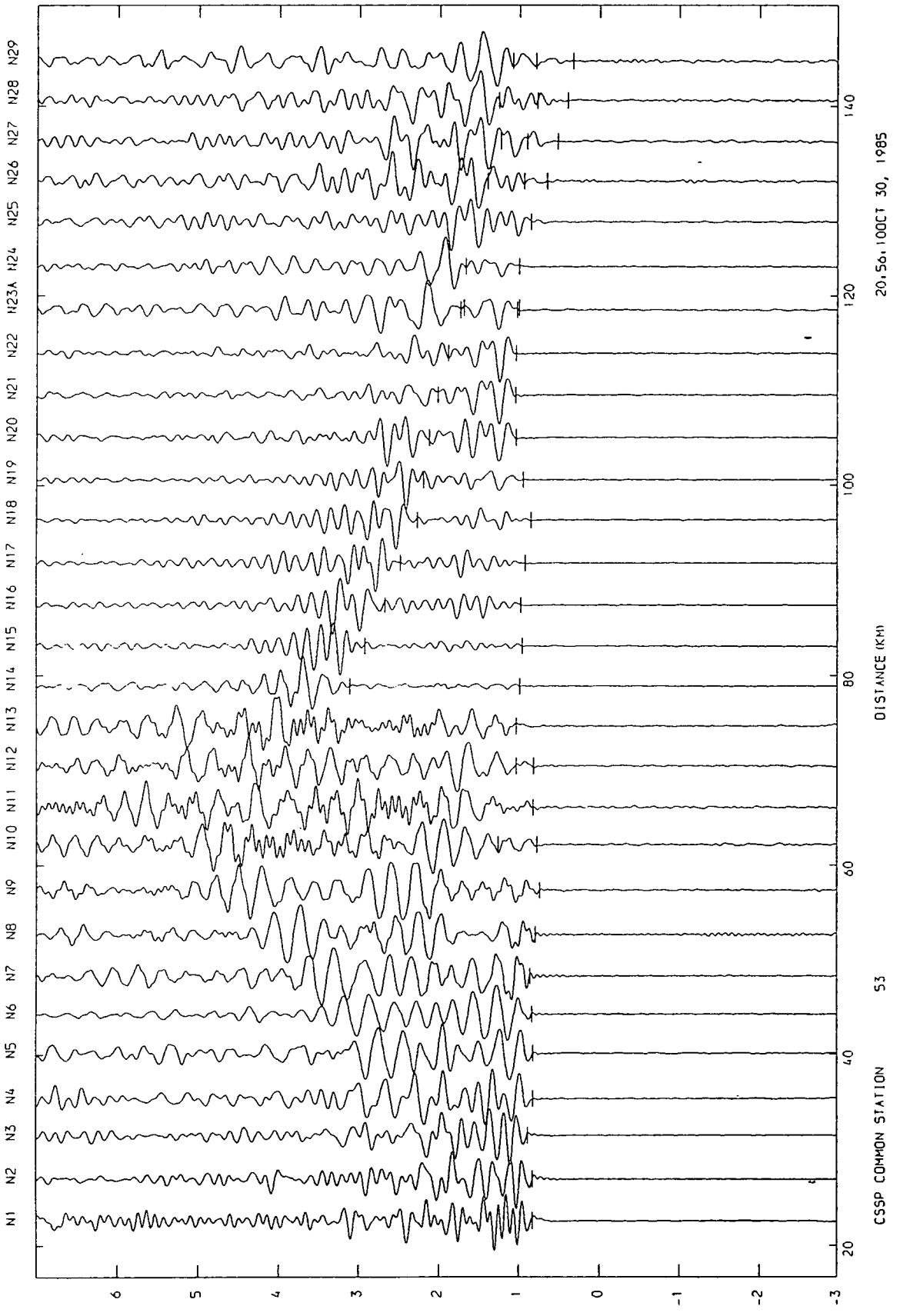
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



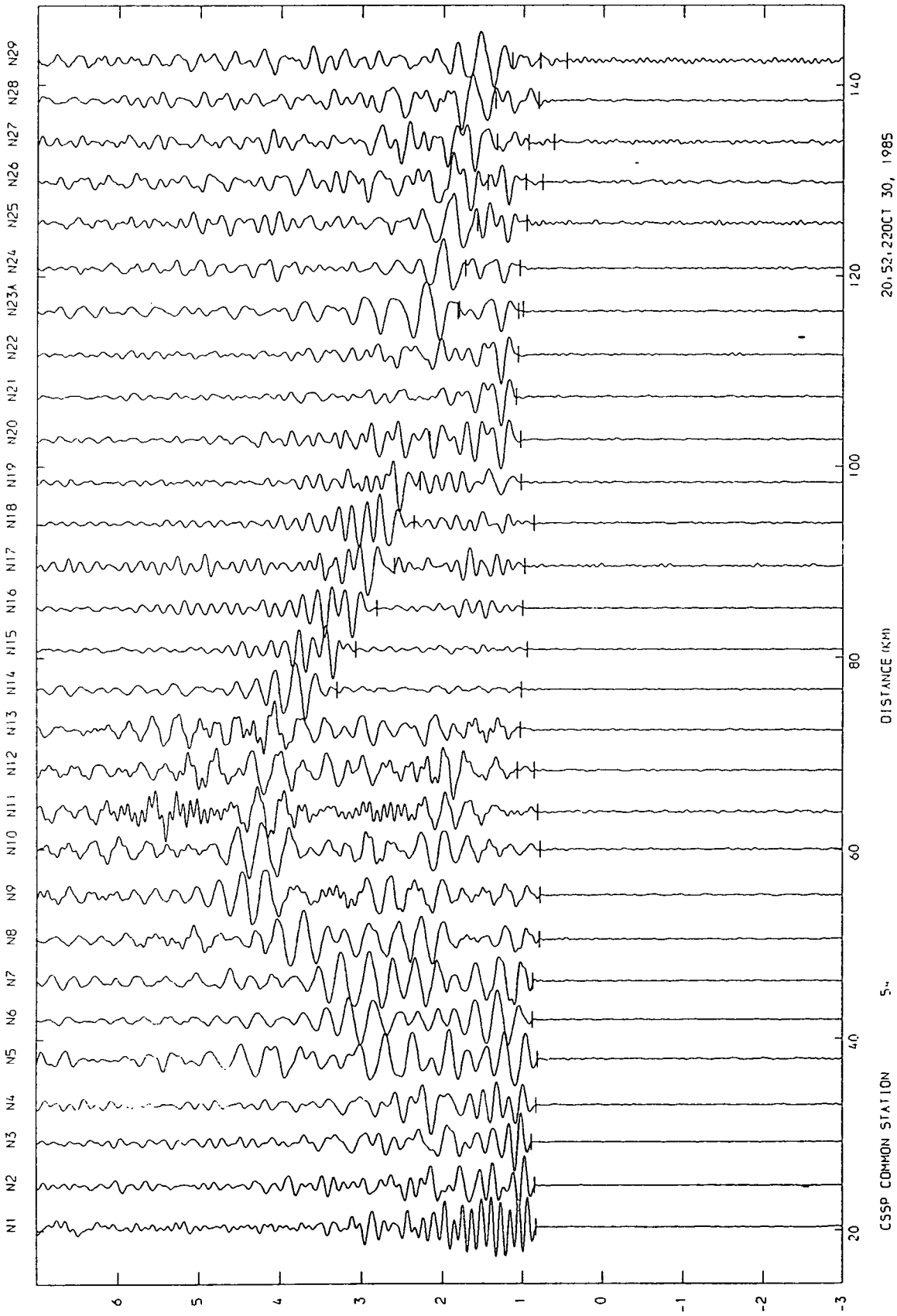
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



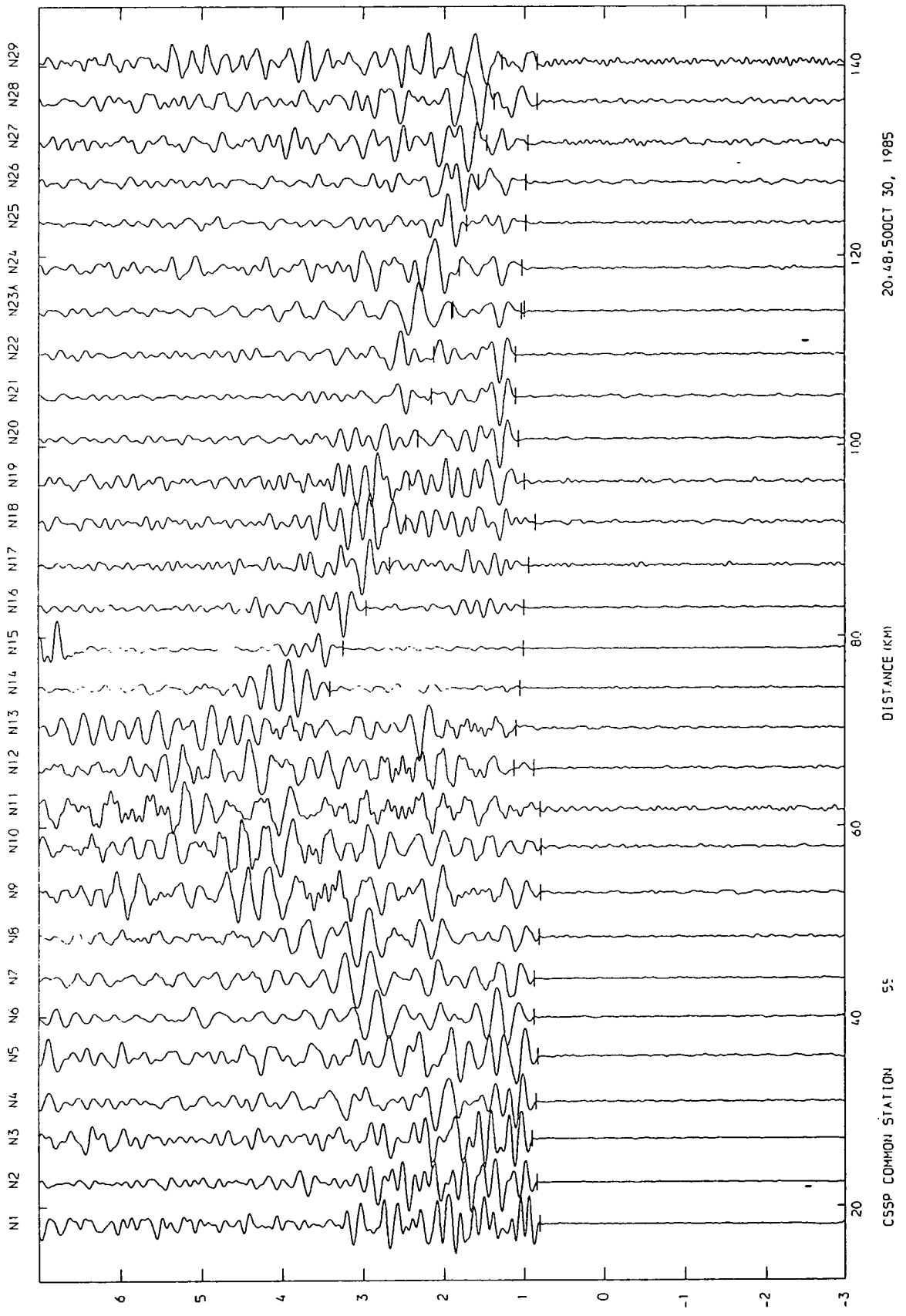
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



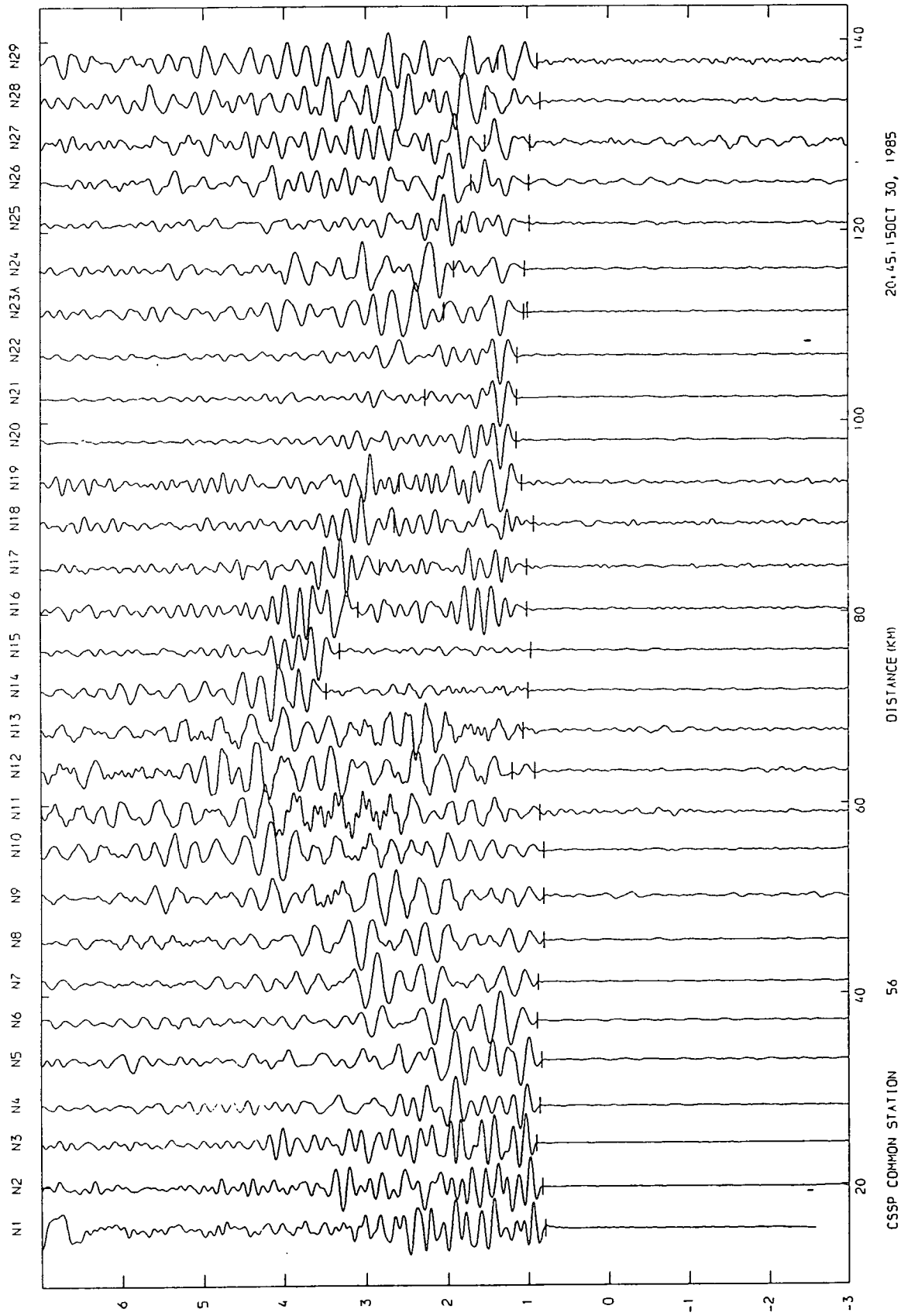
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



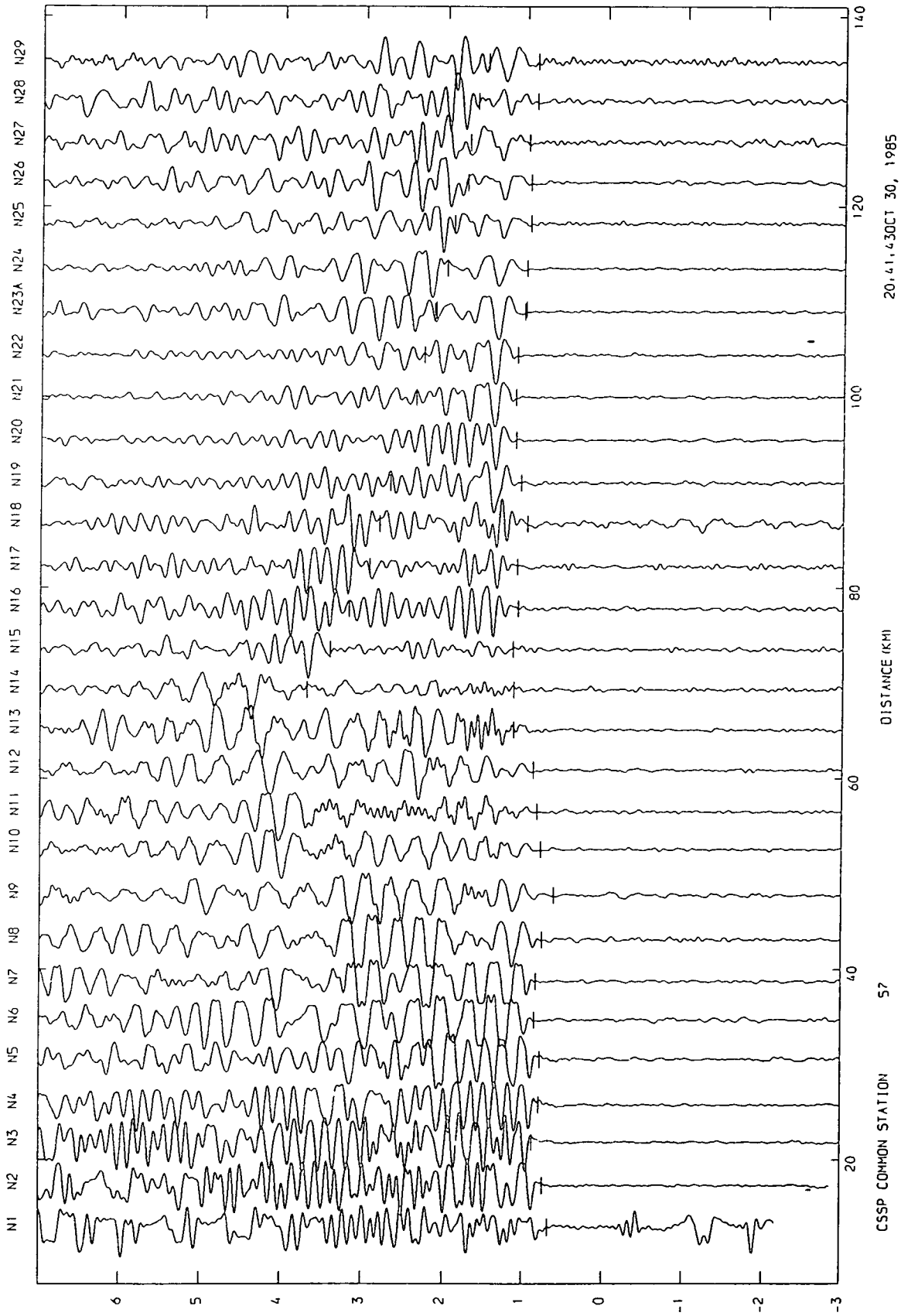
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



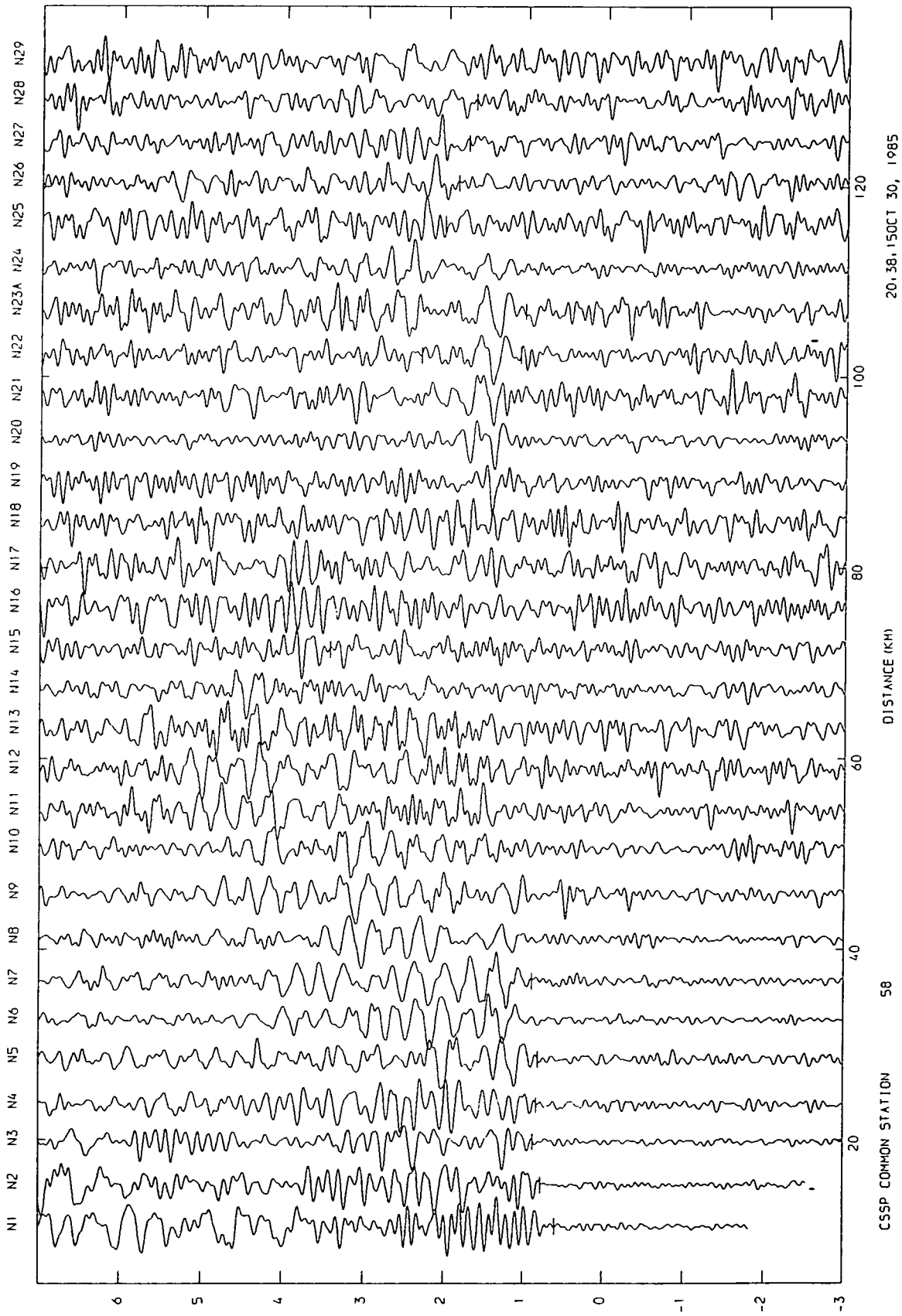
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



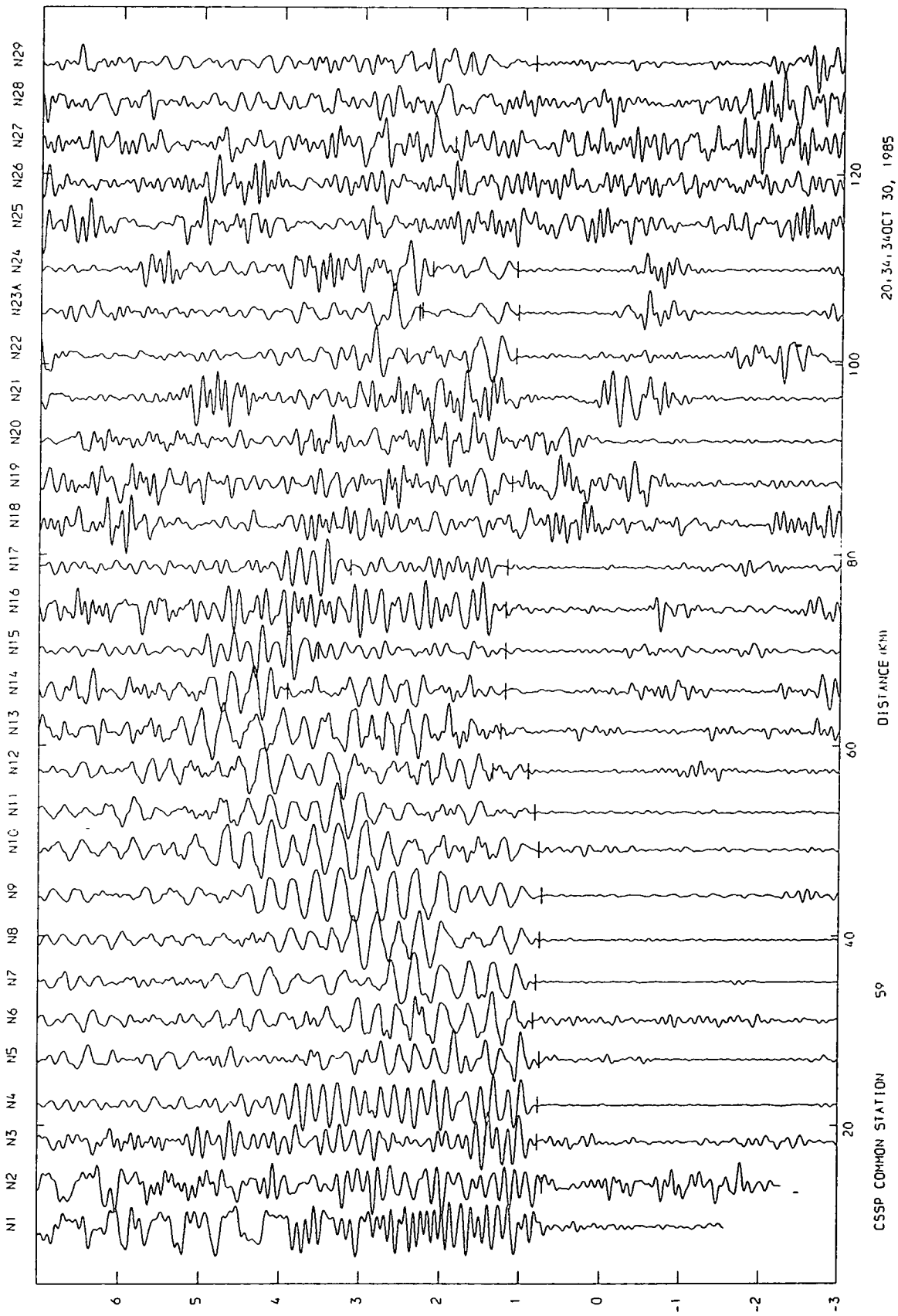
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



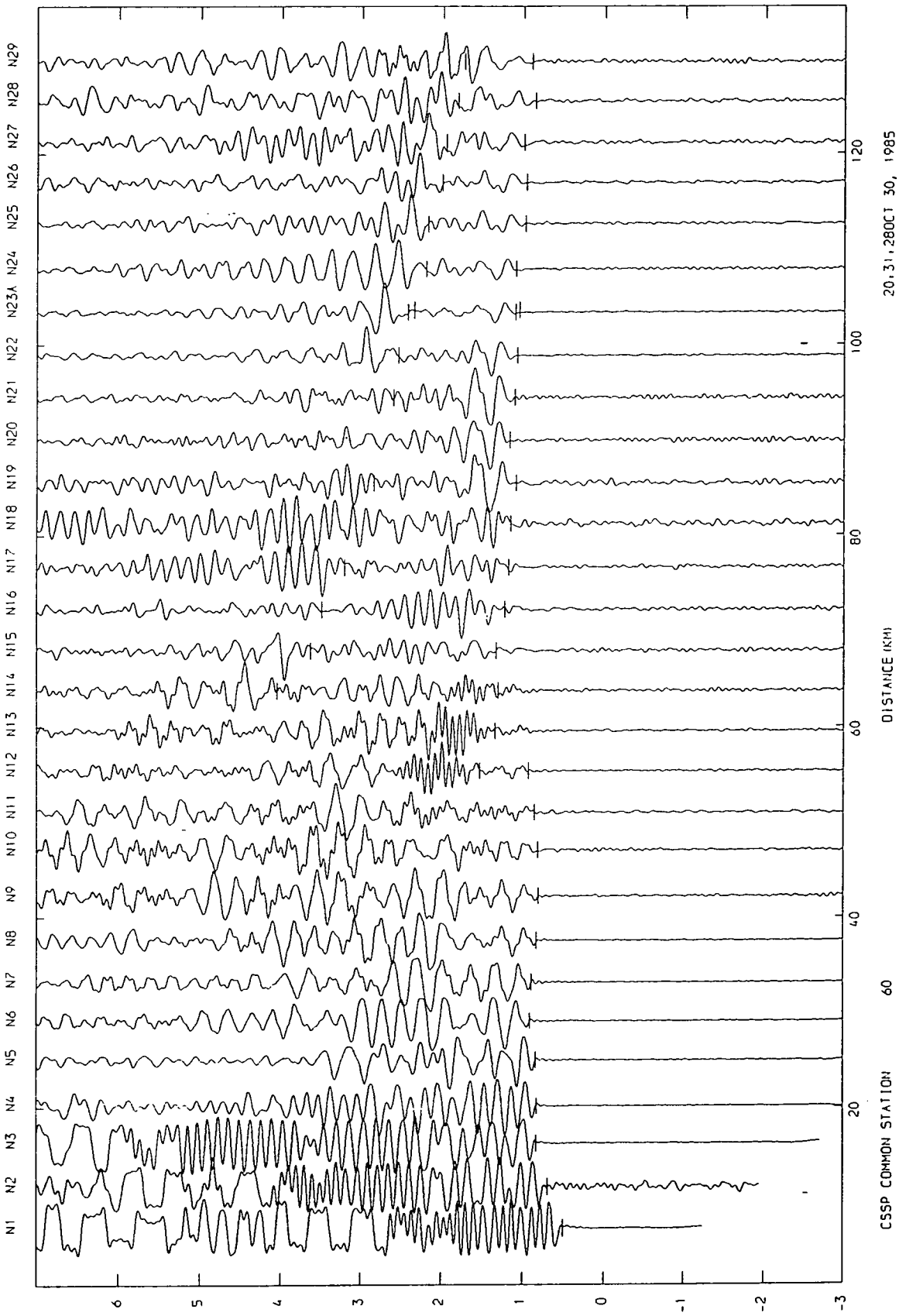
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



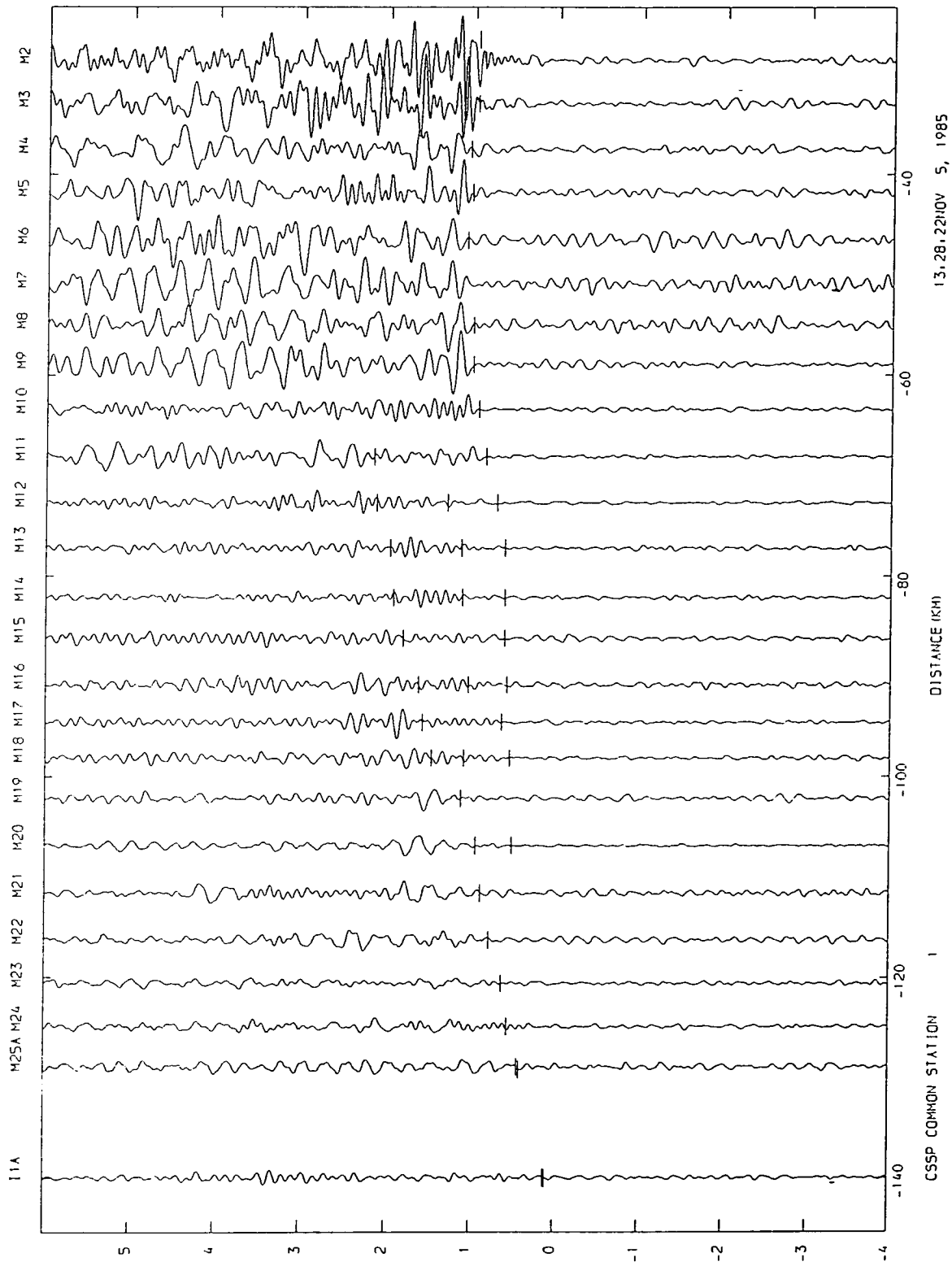
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



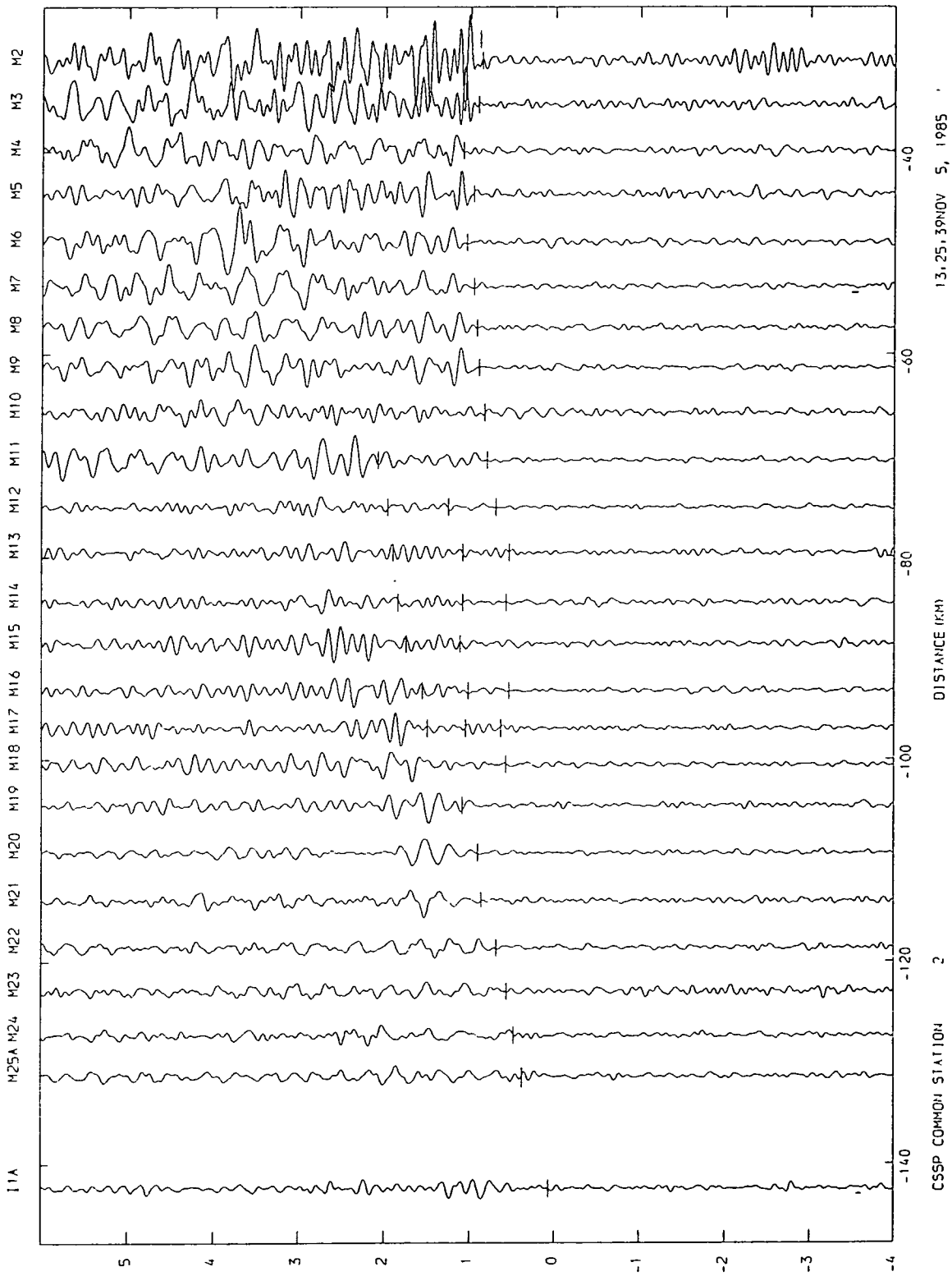
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0021357



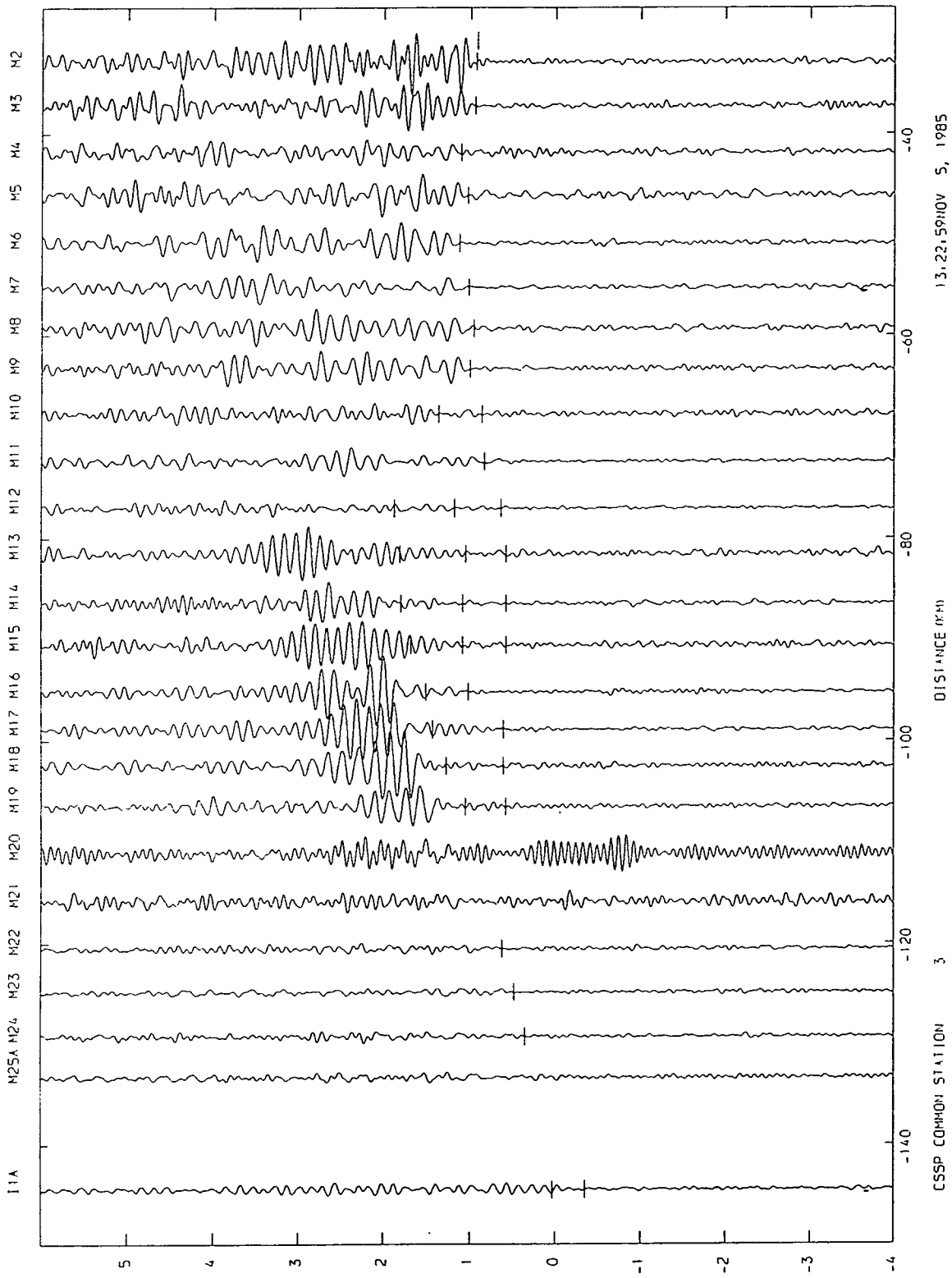
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0024550



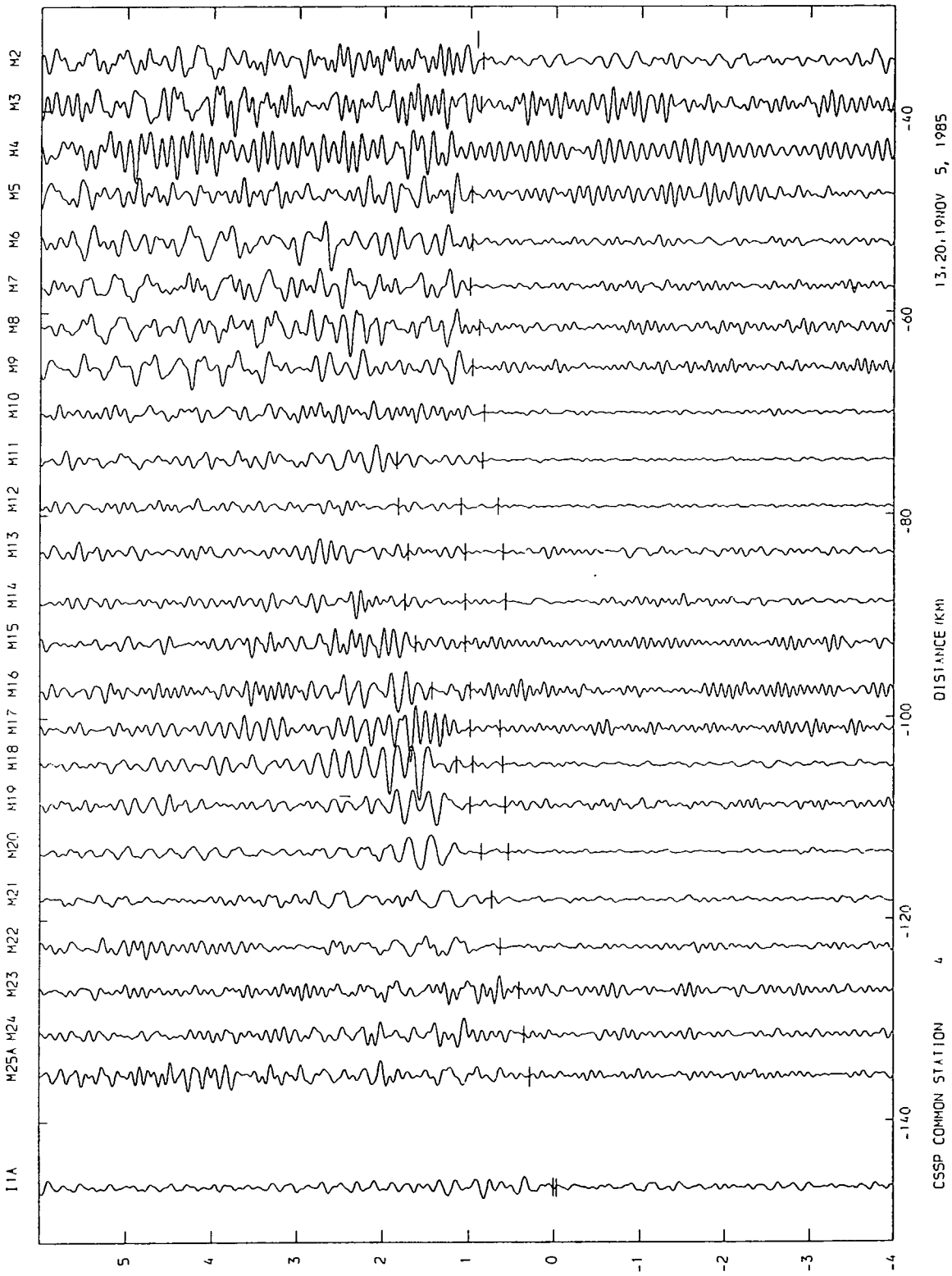
13.25.39NOV 5, 1985

CSSP COMMON STATION 2

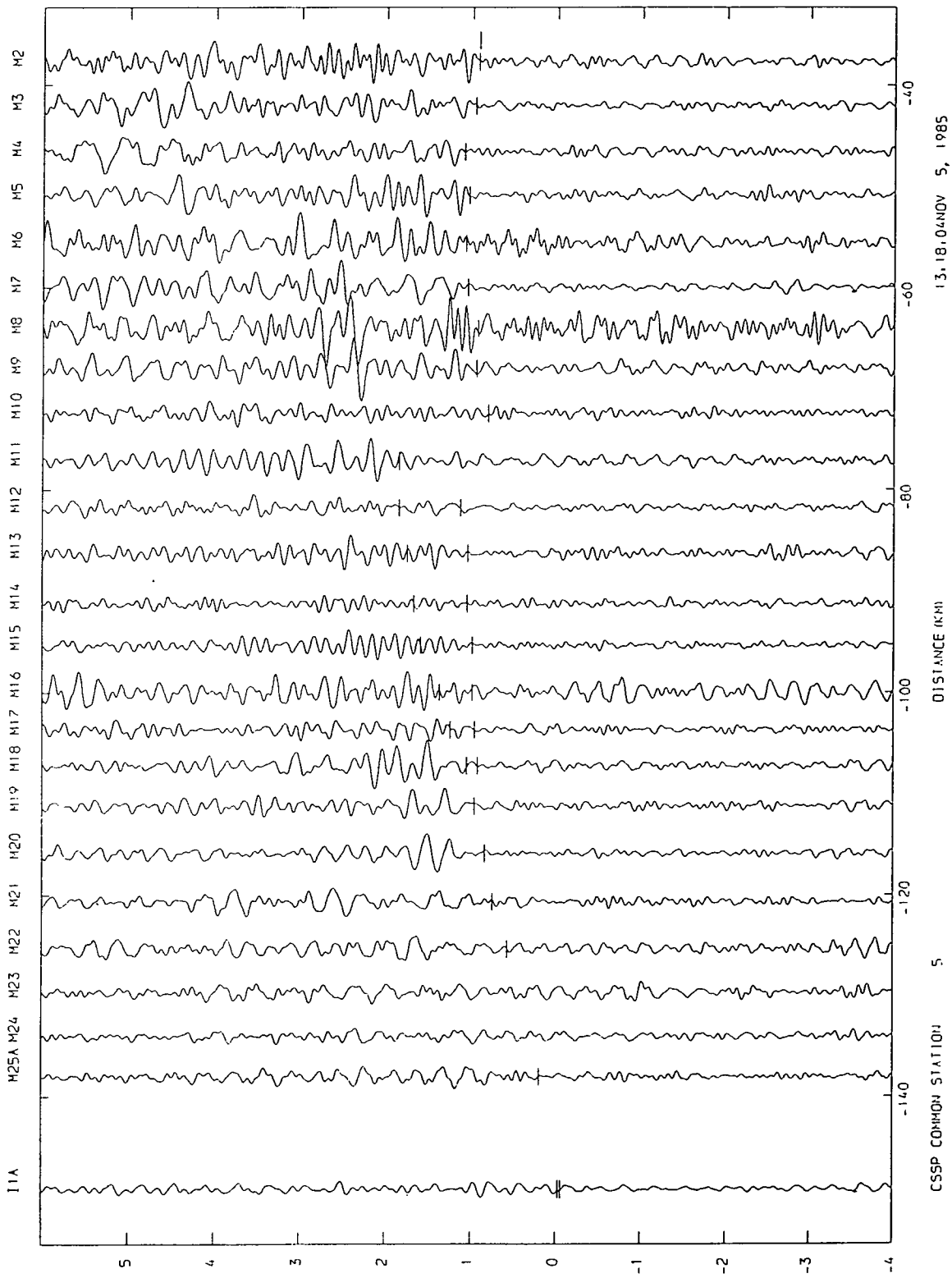
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0007727



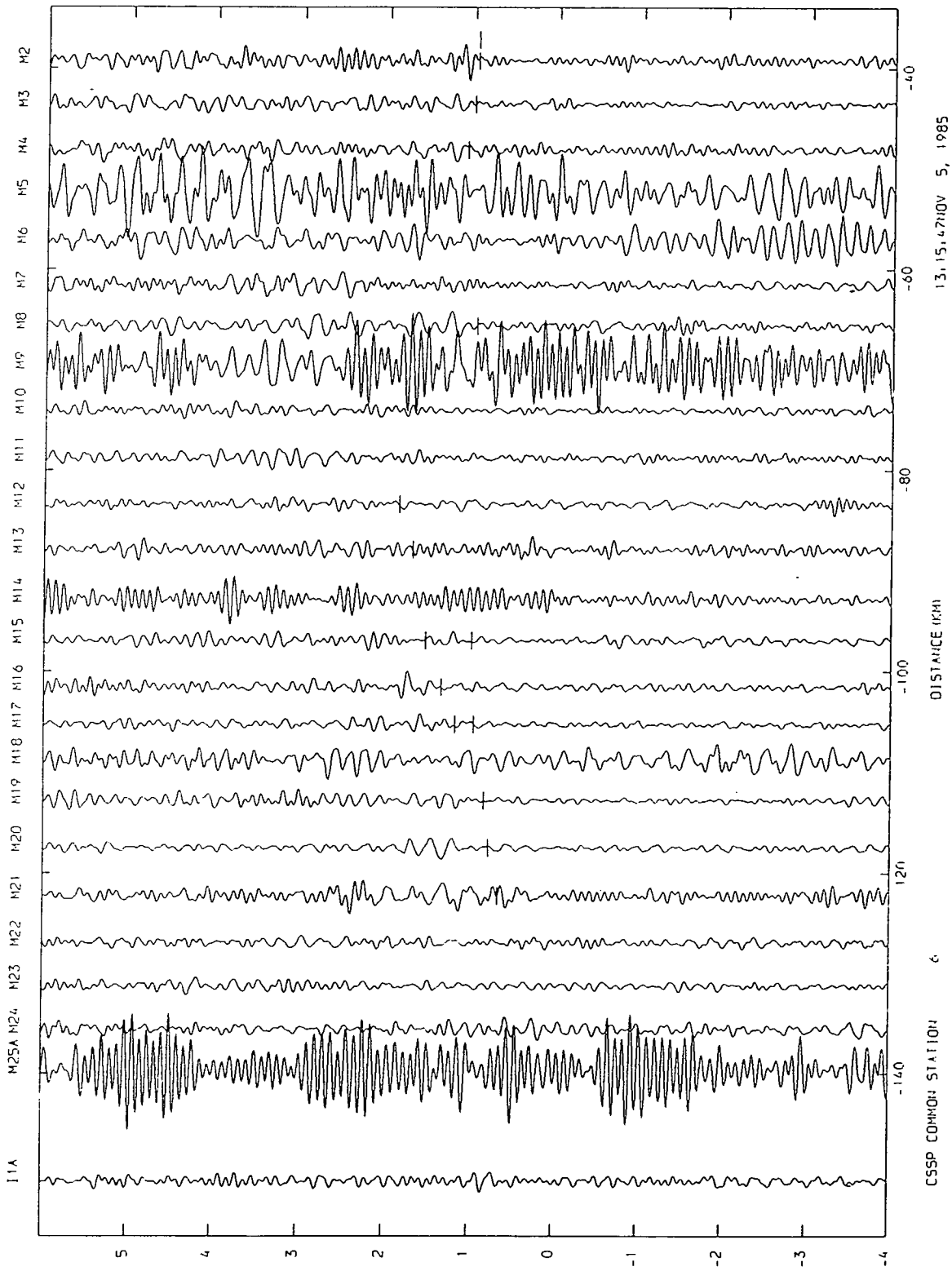
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0004199



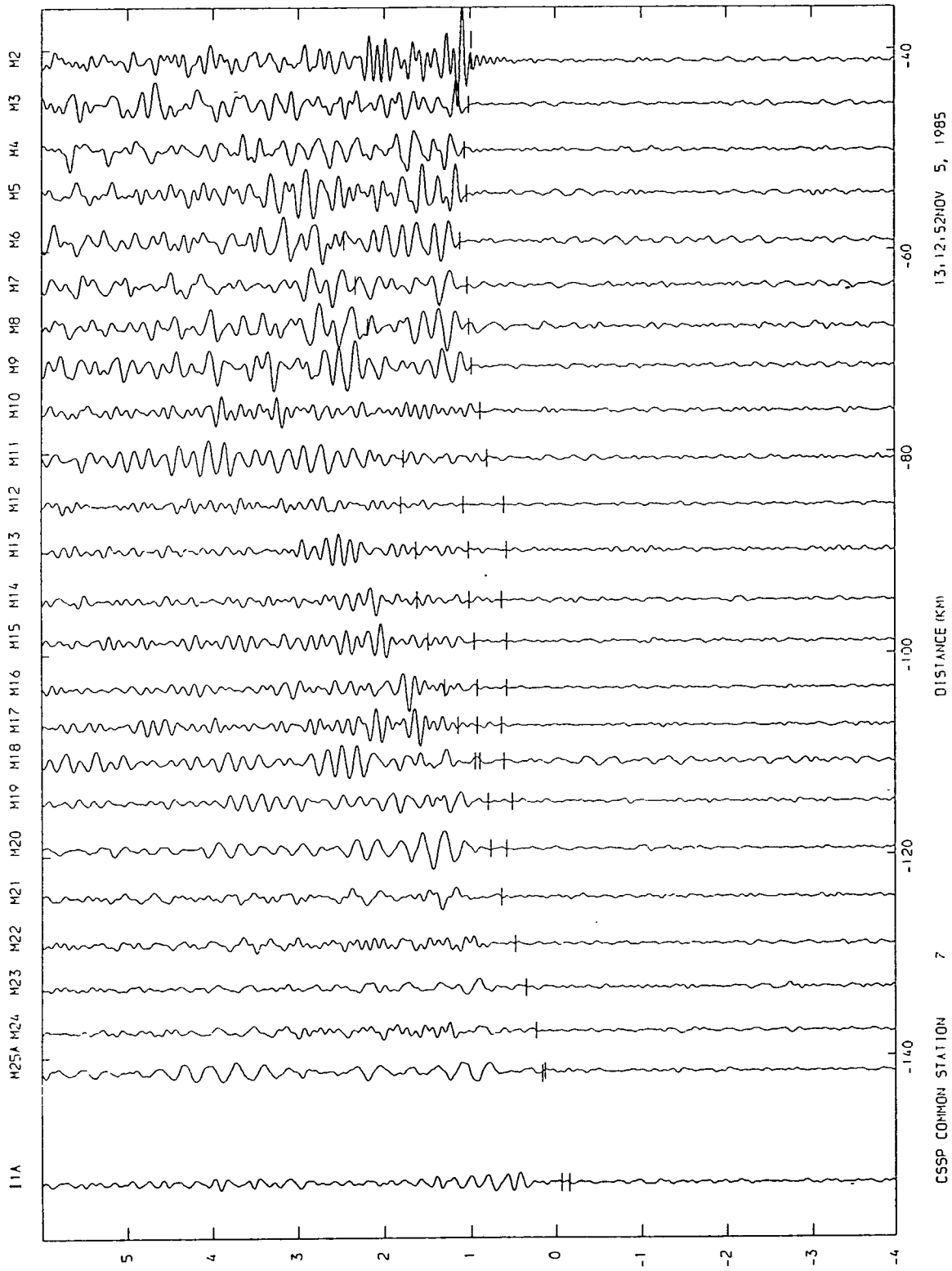
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0021655



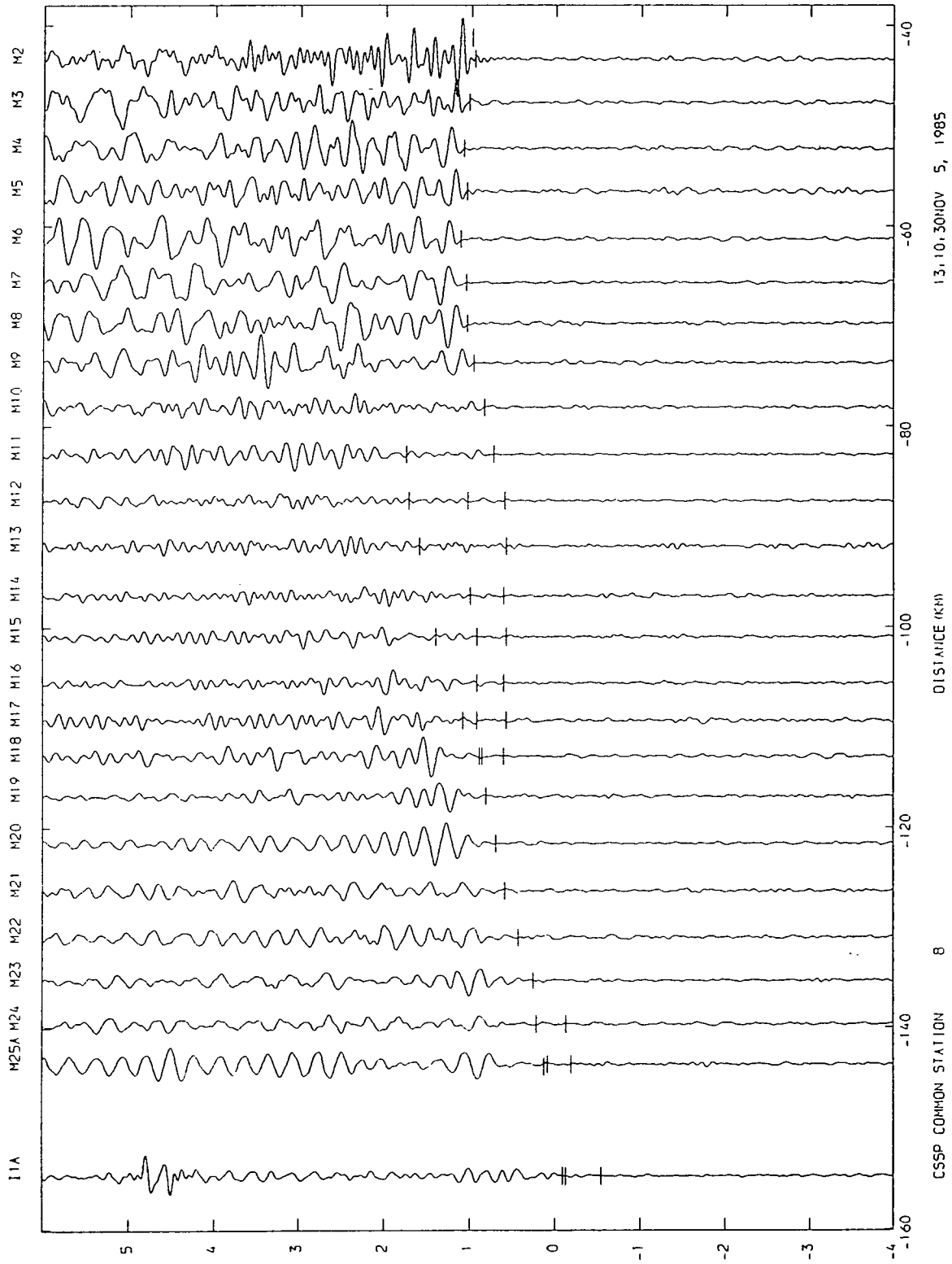
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0012455



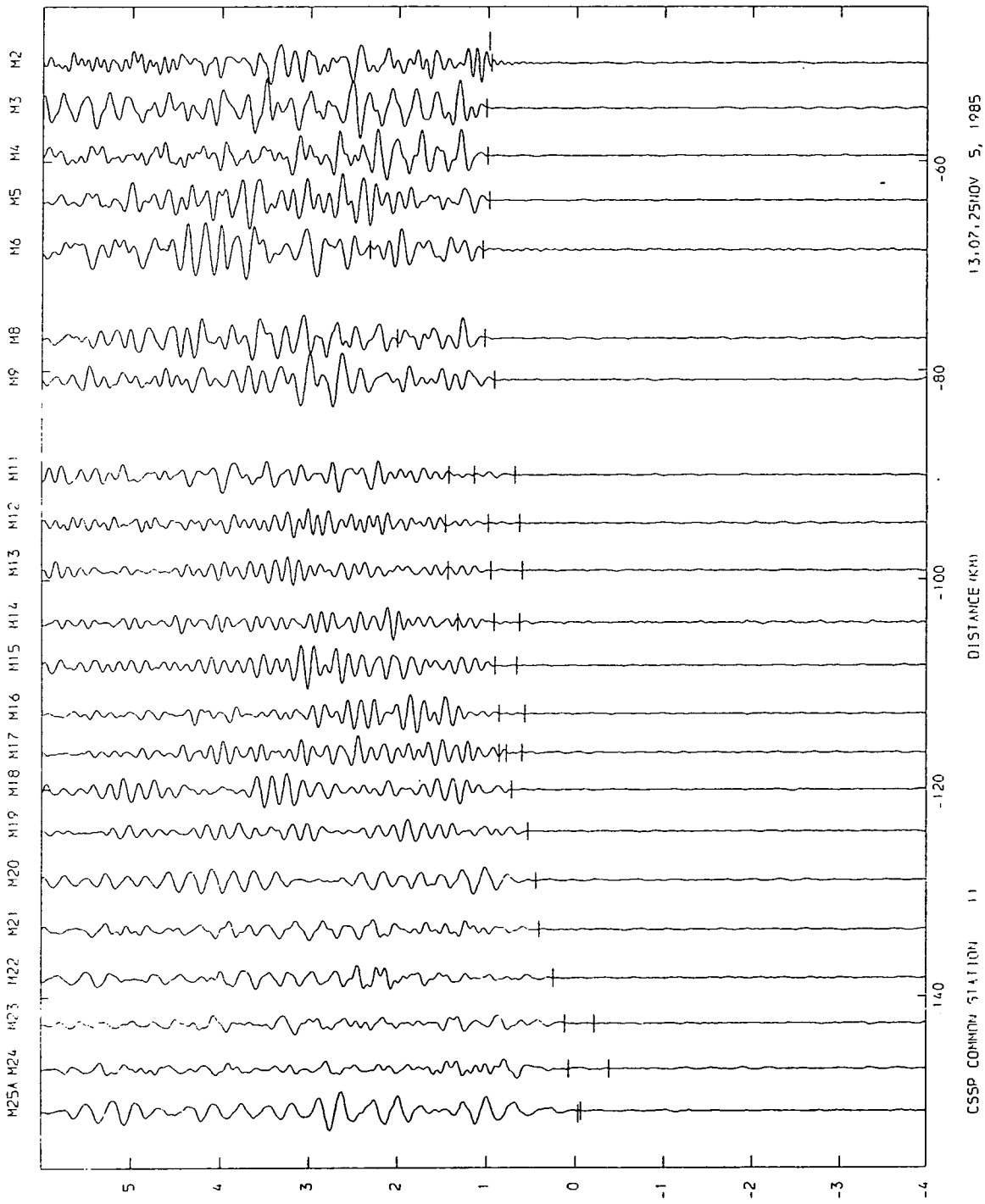
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0022919



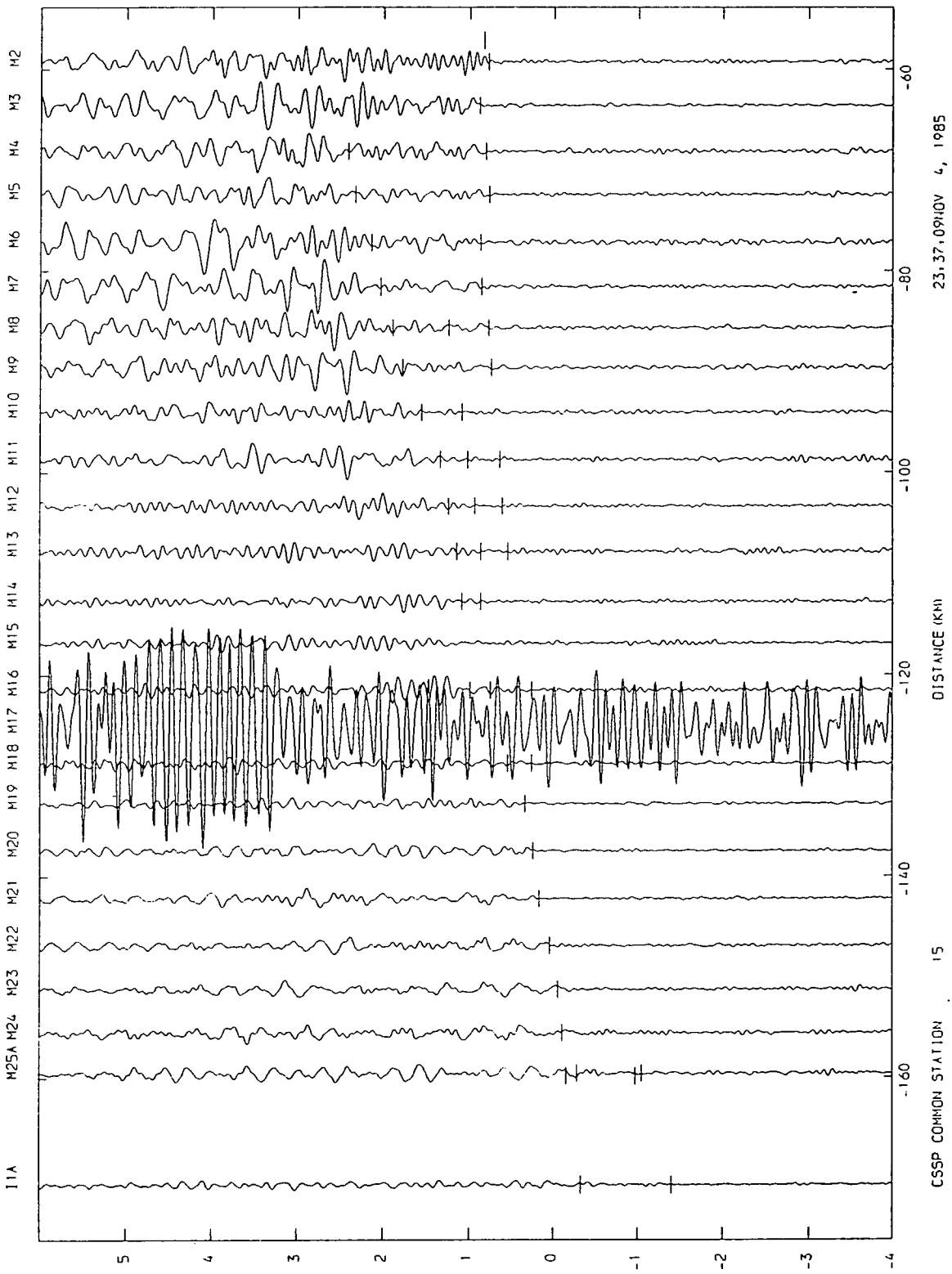
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0031611



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0000015



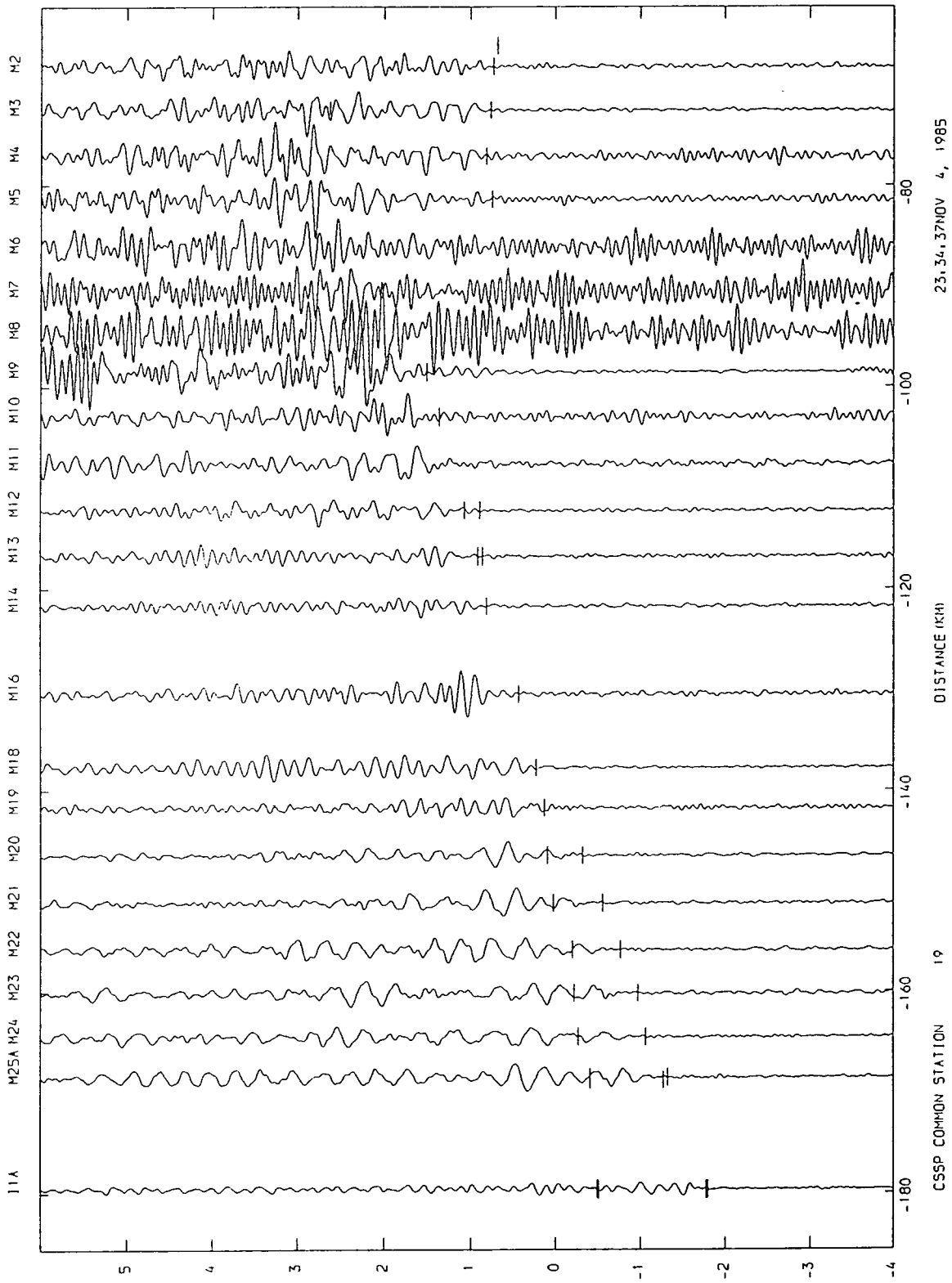
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0042384



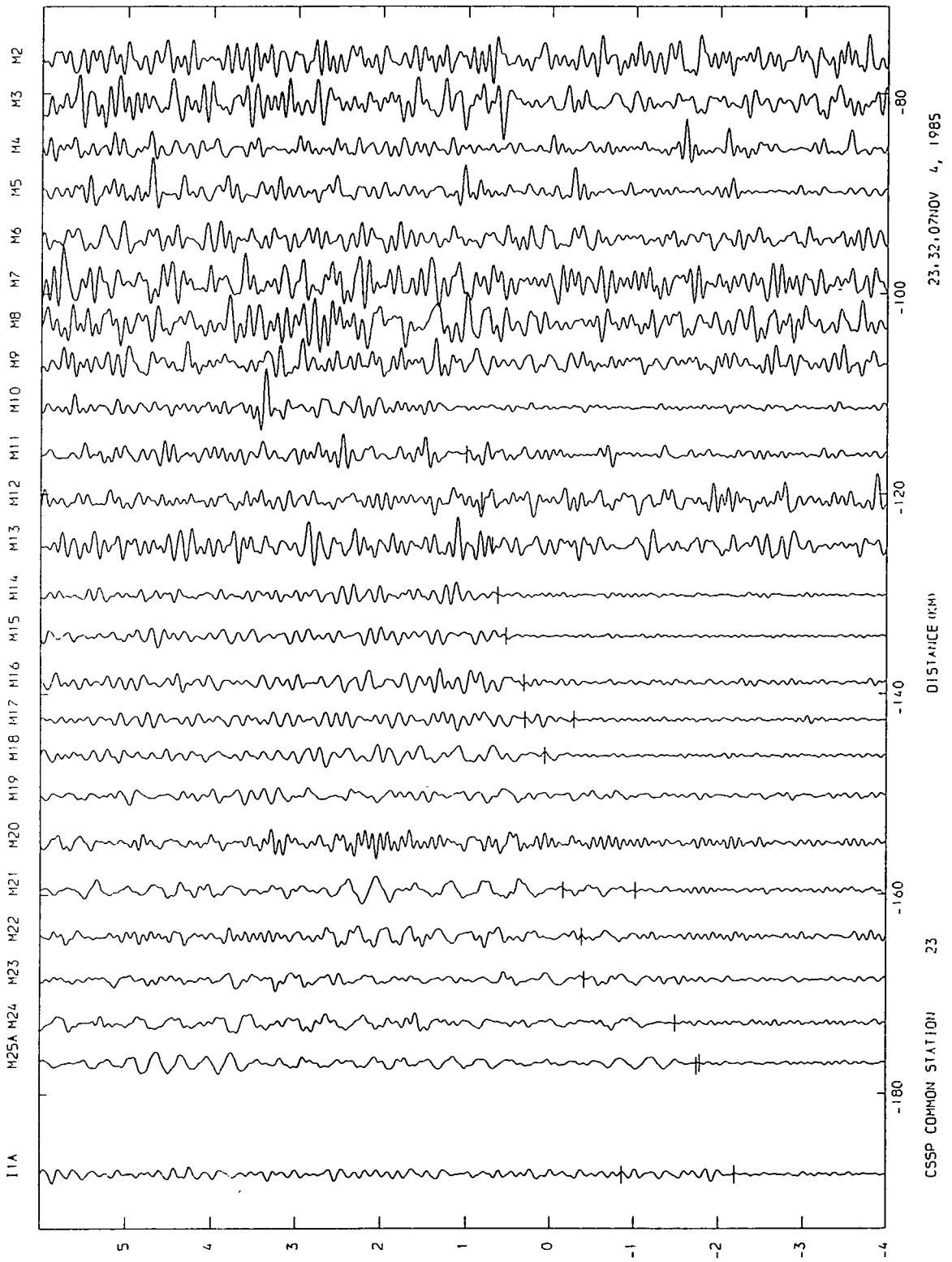
23.37.09NOV 4, 1985

CSSP COMMON STATION 15

FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0038274



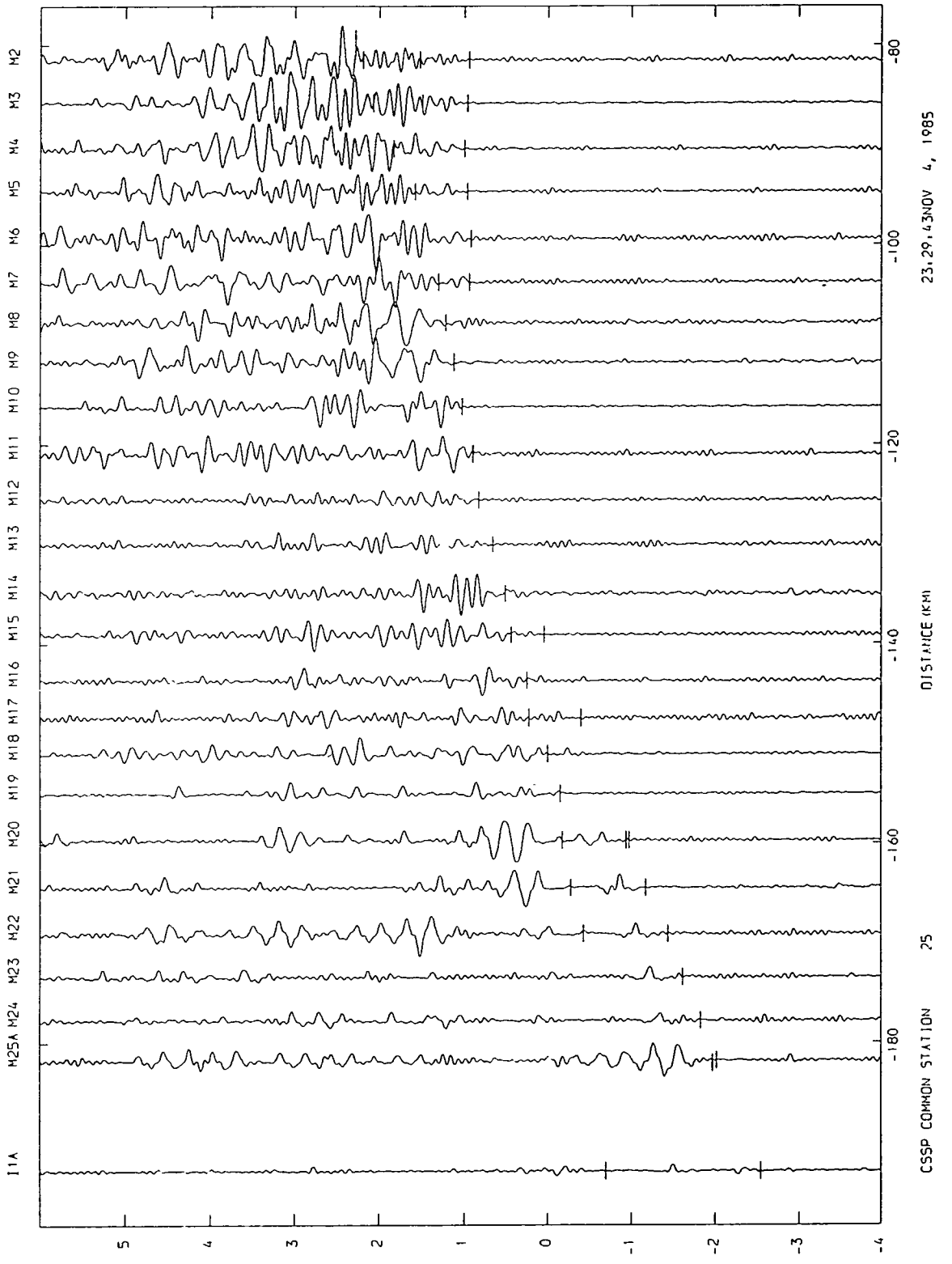
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0039007



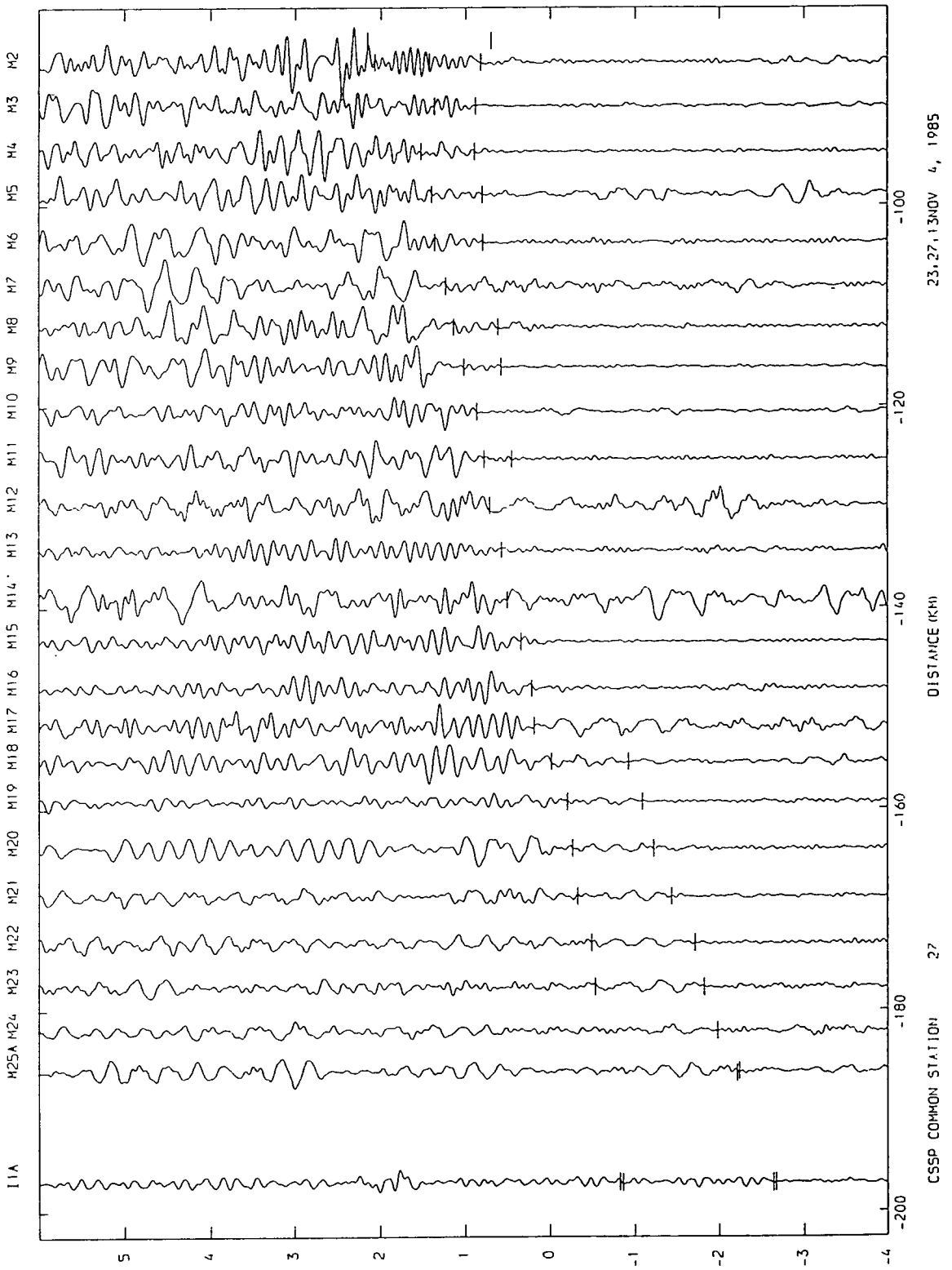
23:32:07NOV 4, 1985

CSSP COMMON STATION 23

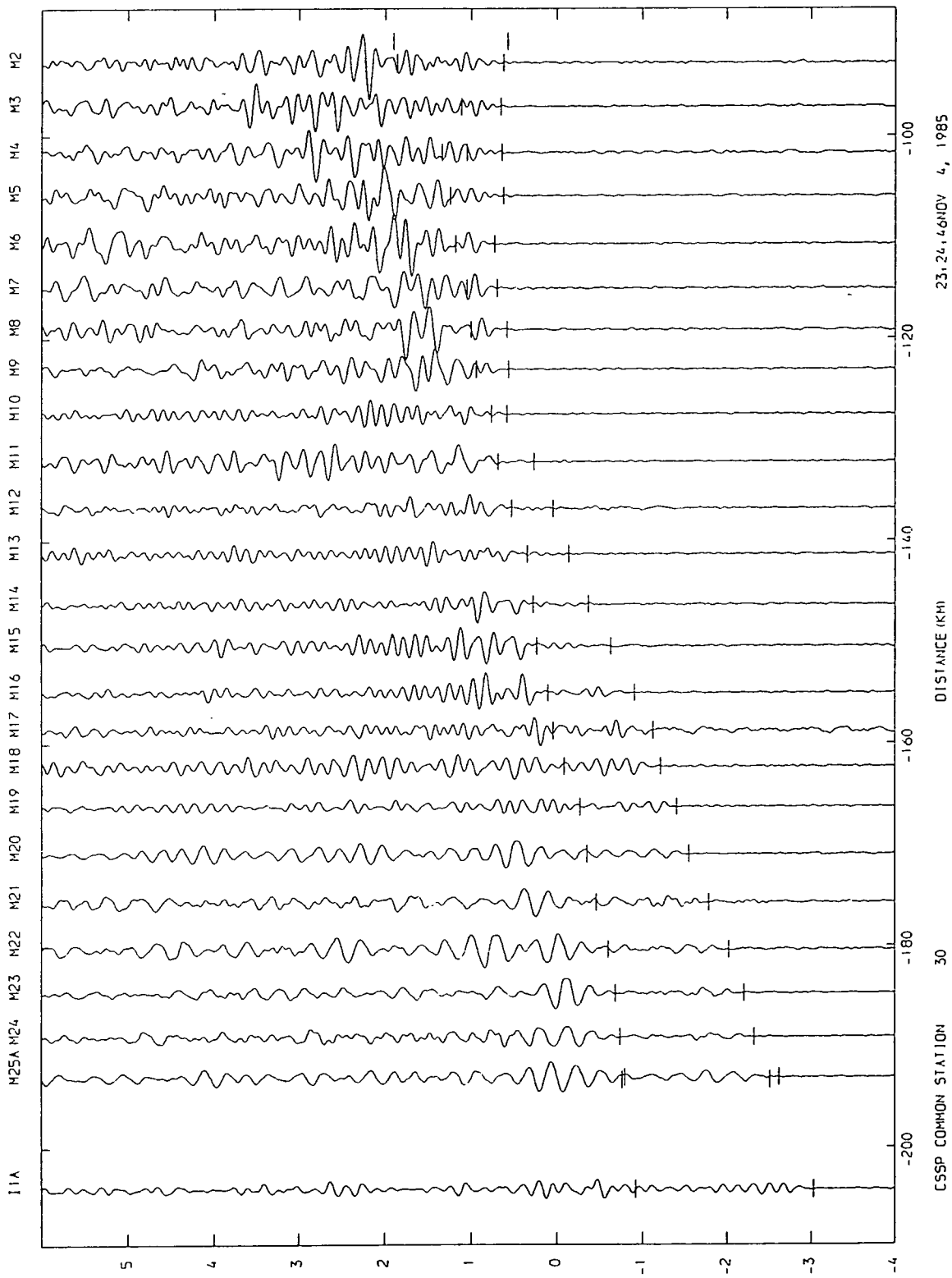
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0059451



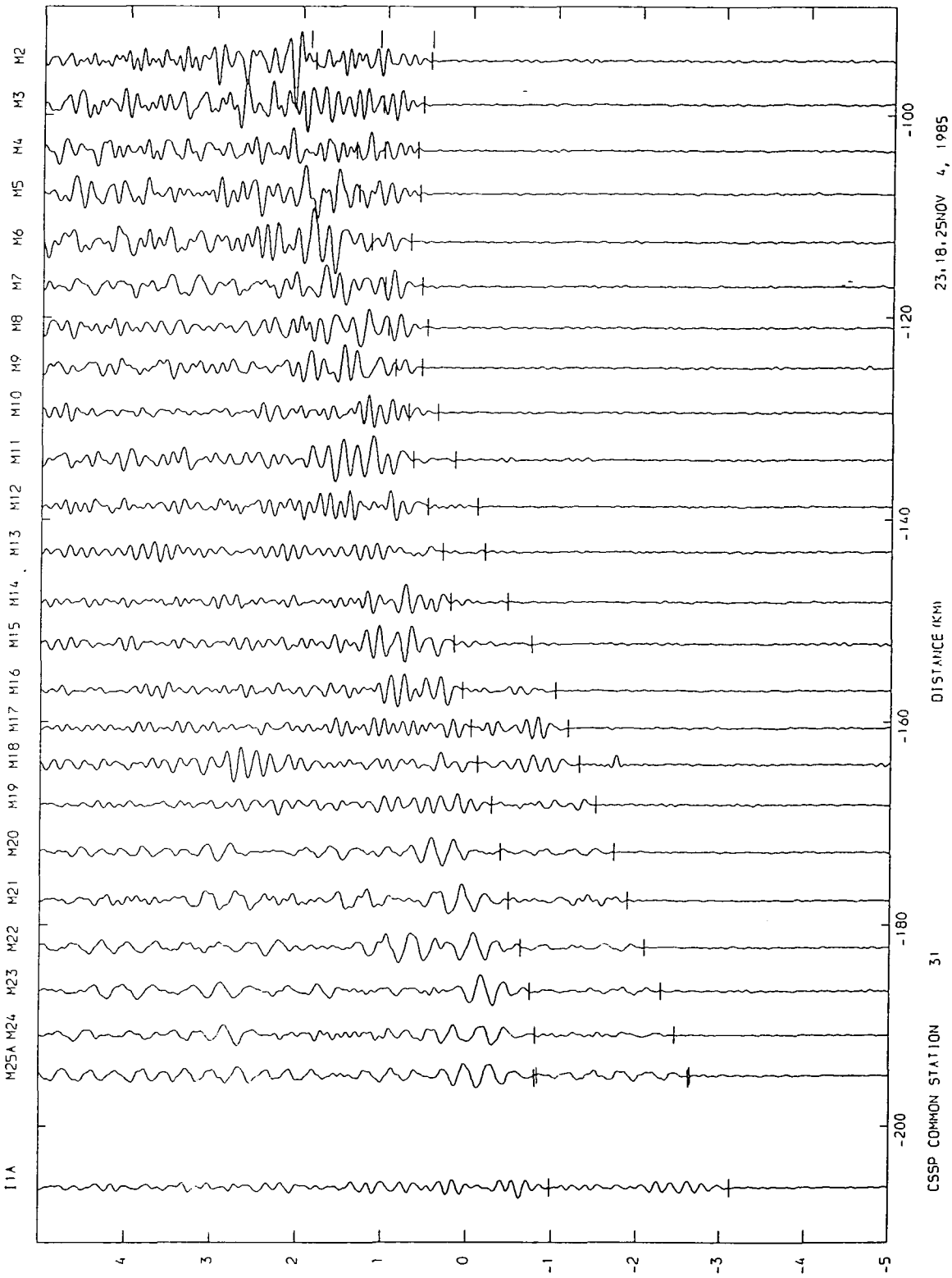
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0078527



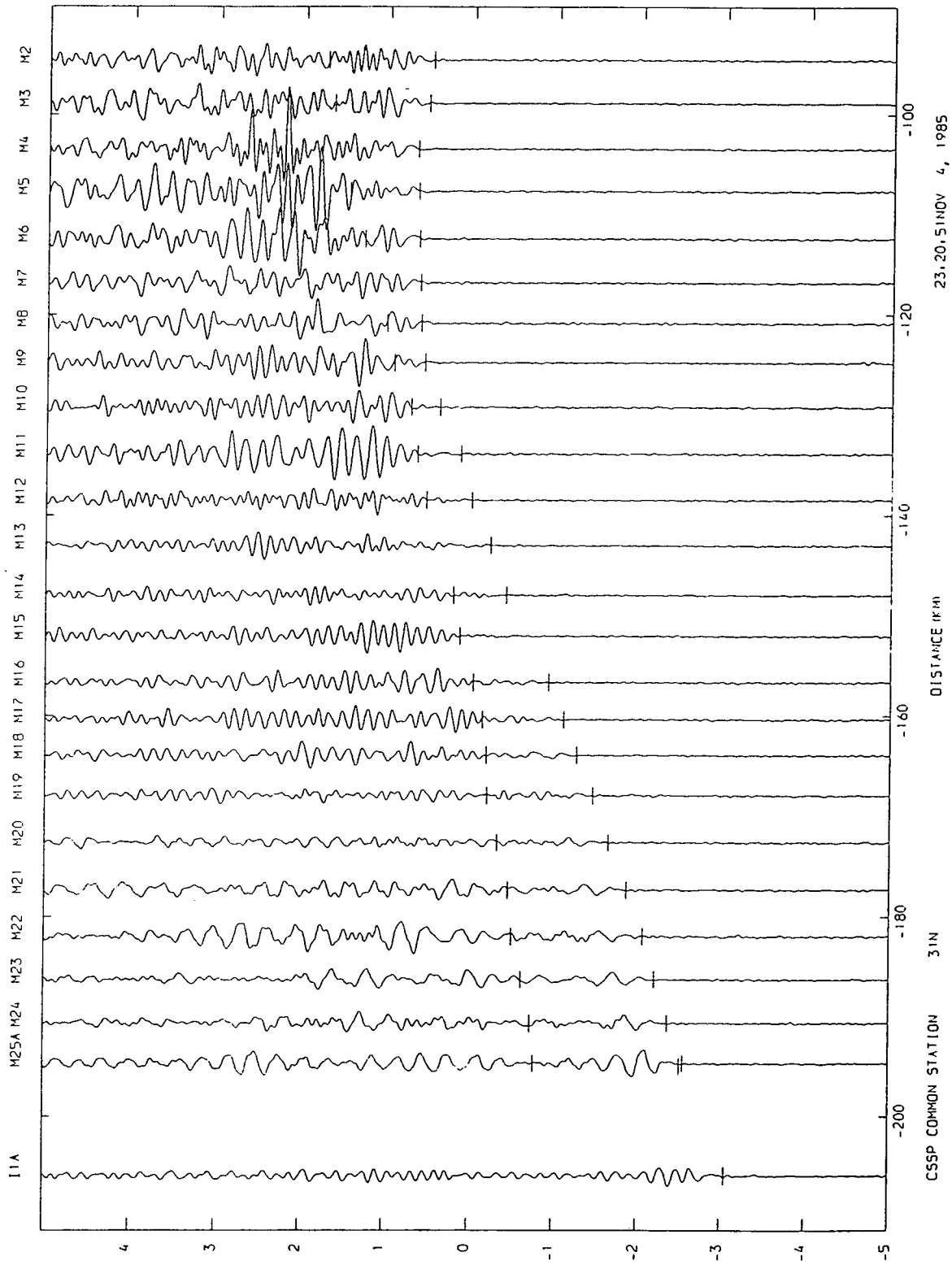
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0041259



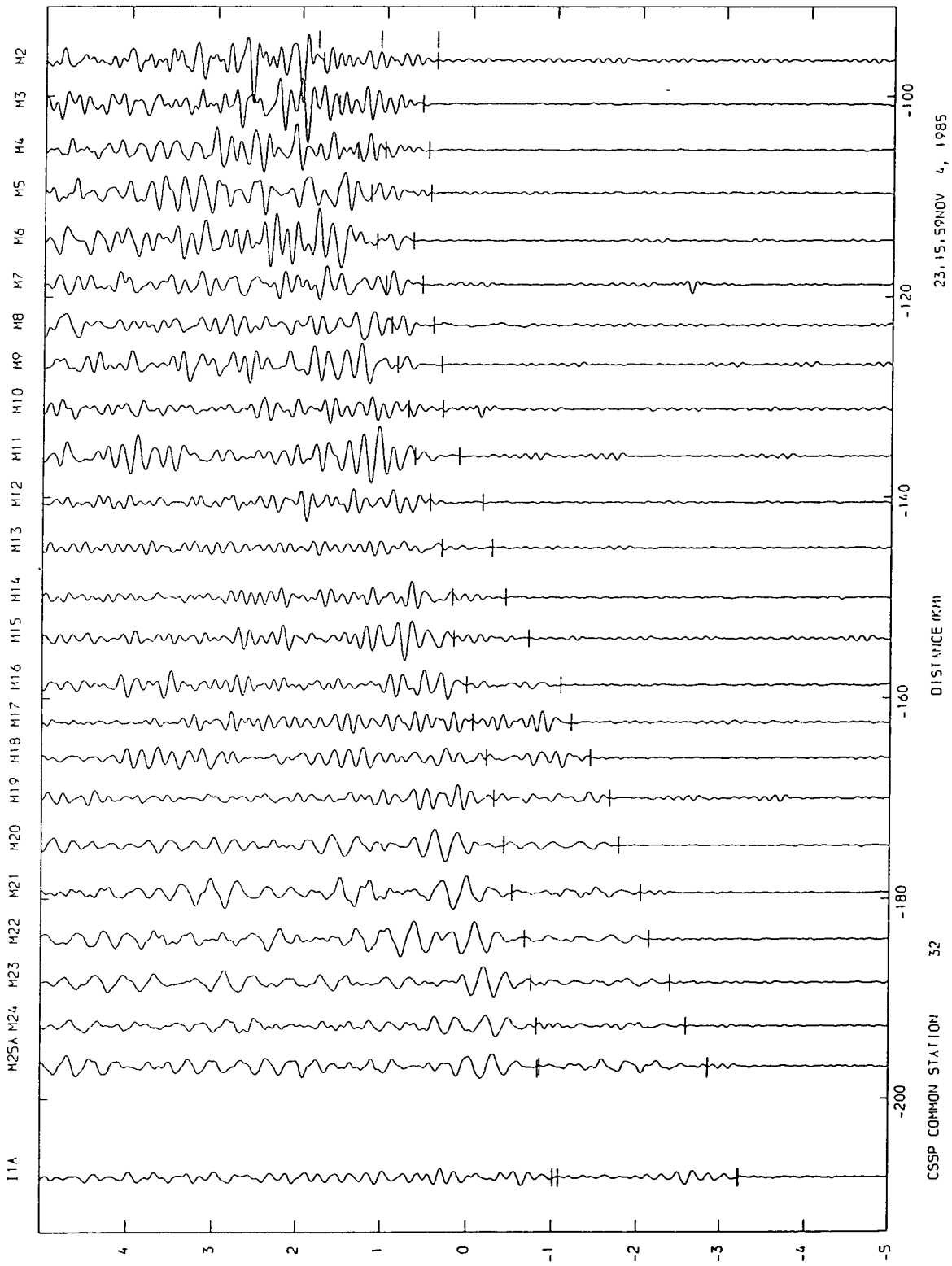
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0094053



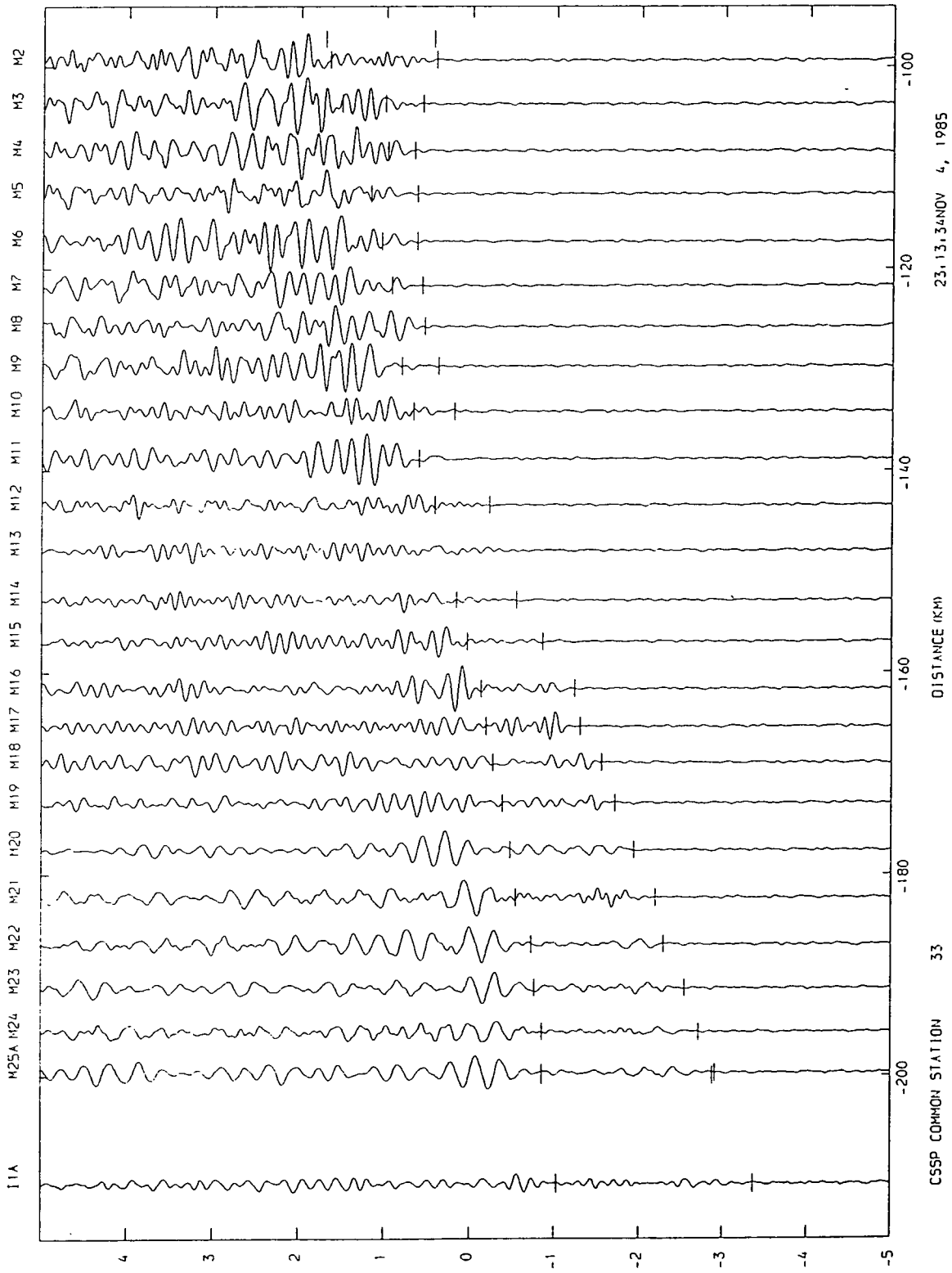
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0054992



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0078700



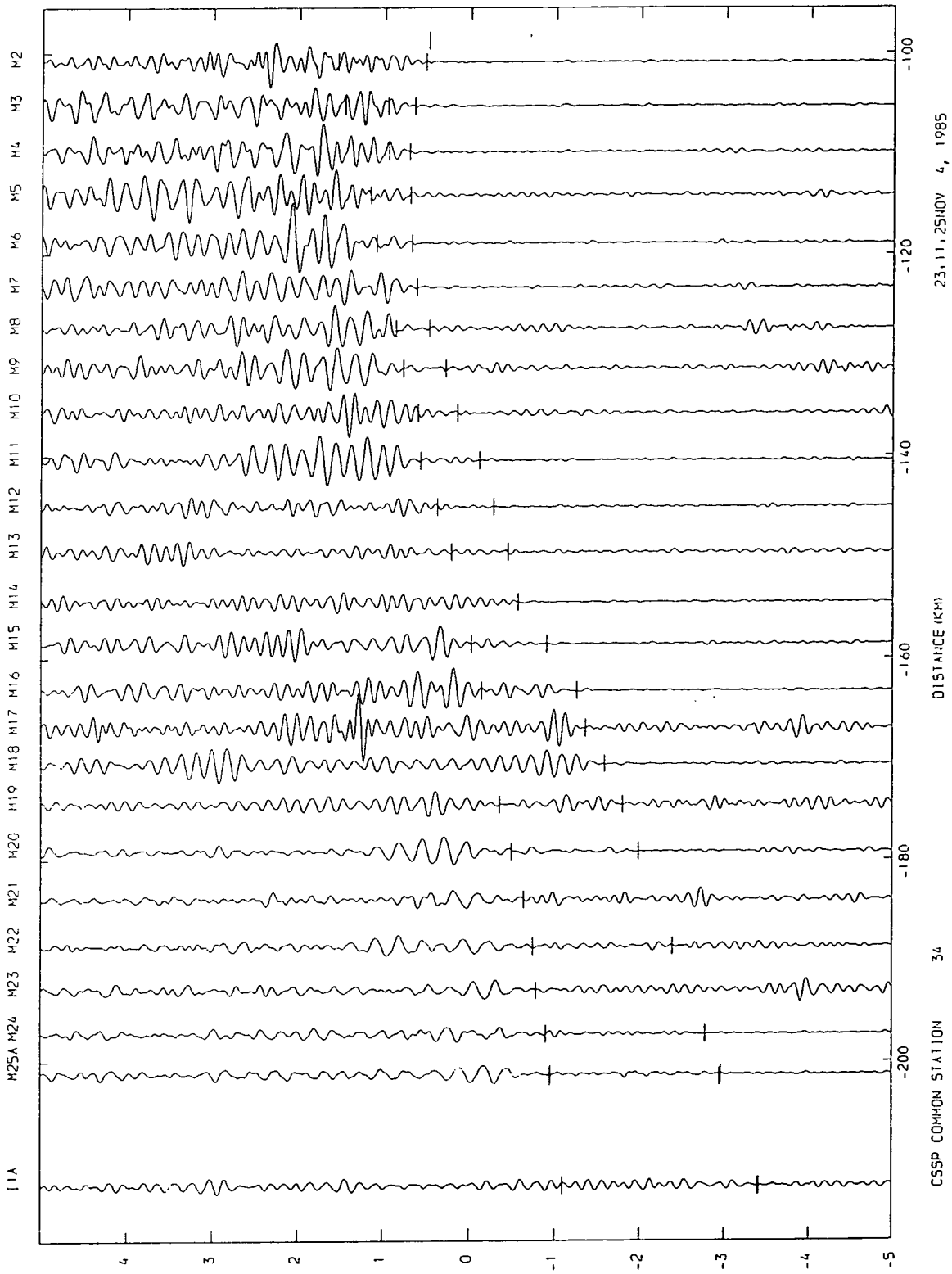
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0464026



CSSP COMMON STATION 33

23:13:34NOV 4, 1985

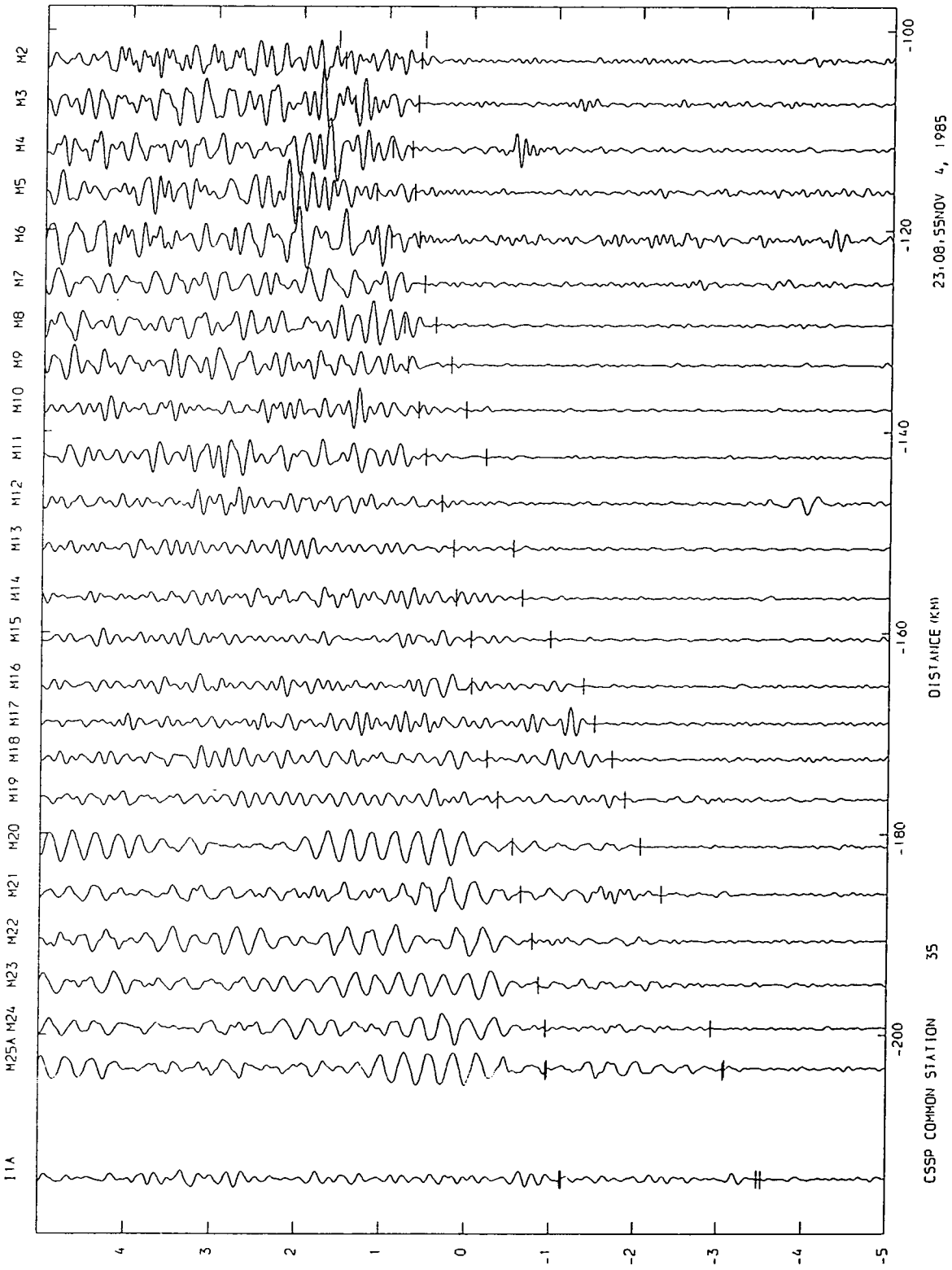
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0054399



23,11,25NOV 4, 1985

CSSP COMMON STATION 34

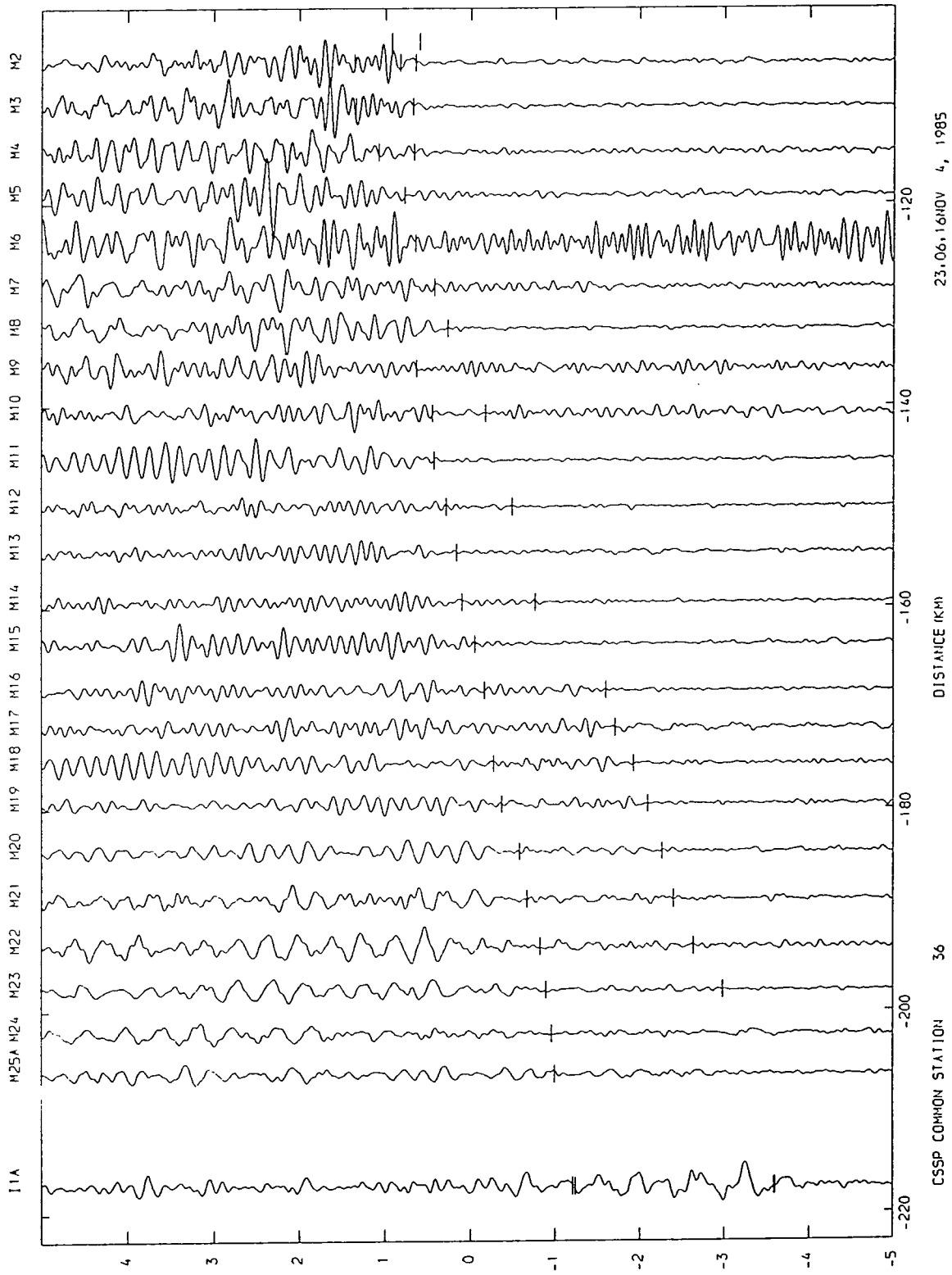
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0064378



23:08:55NOV 4, 1985

CSSP COMMON STATION 35

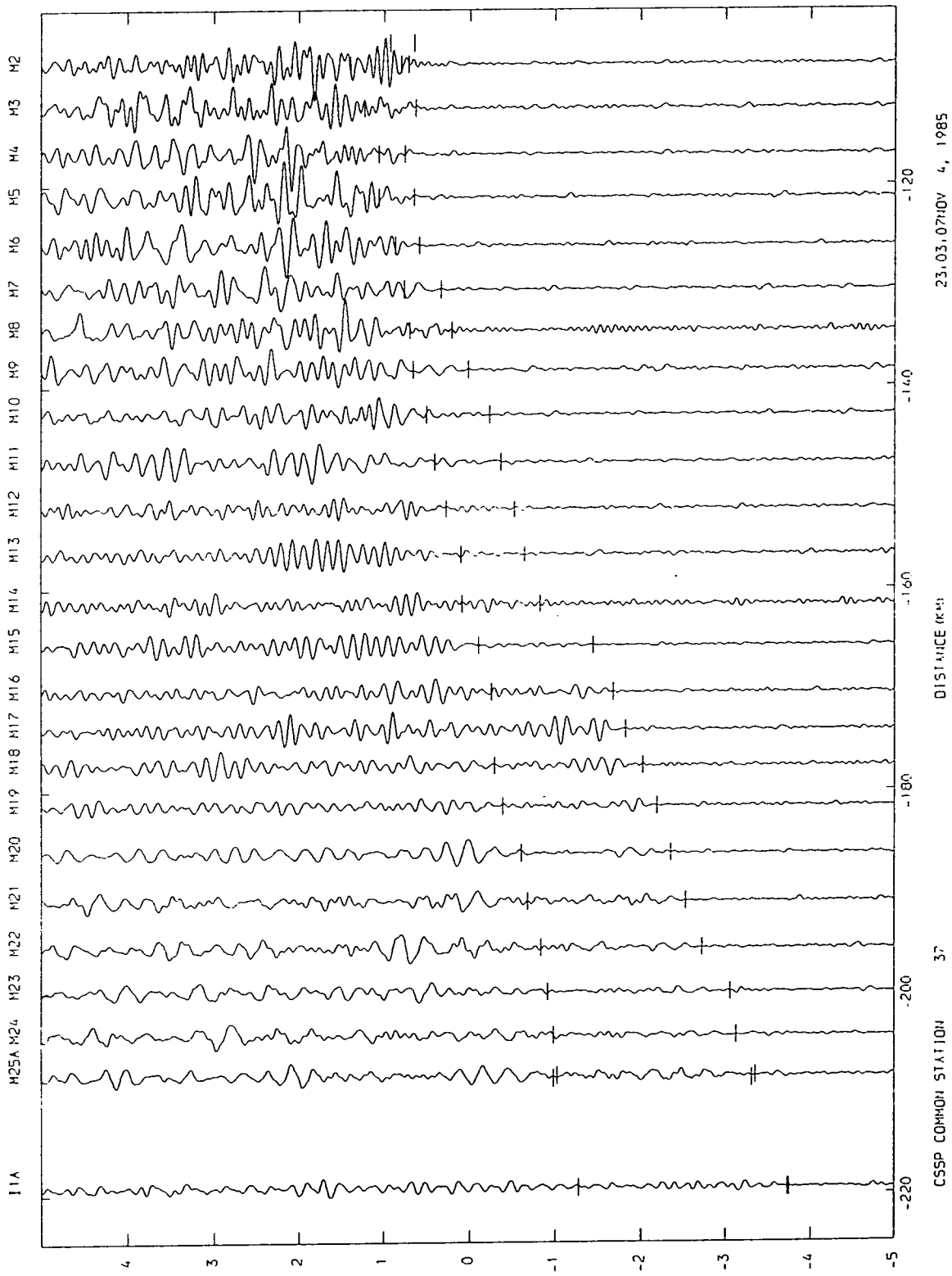
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0085683



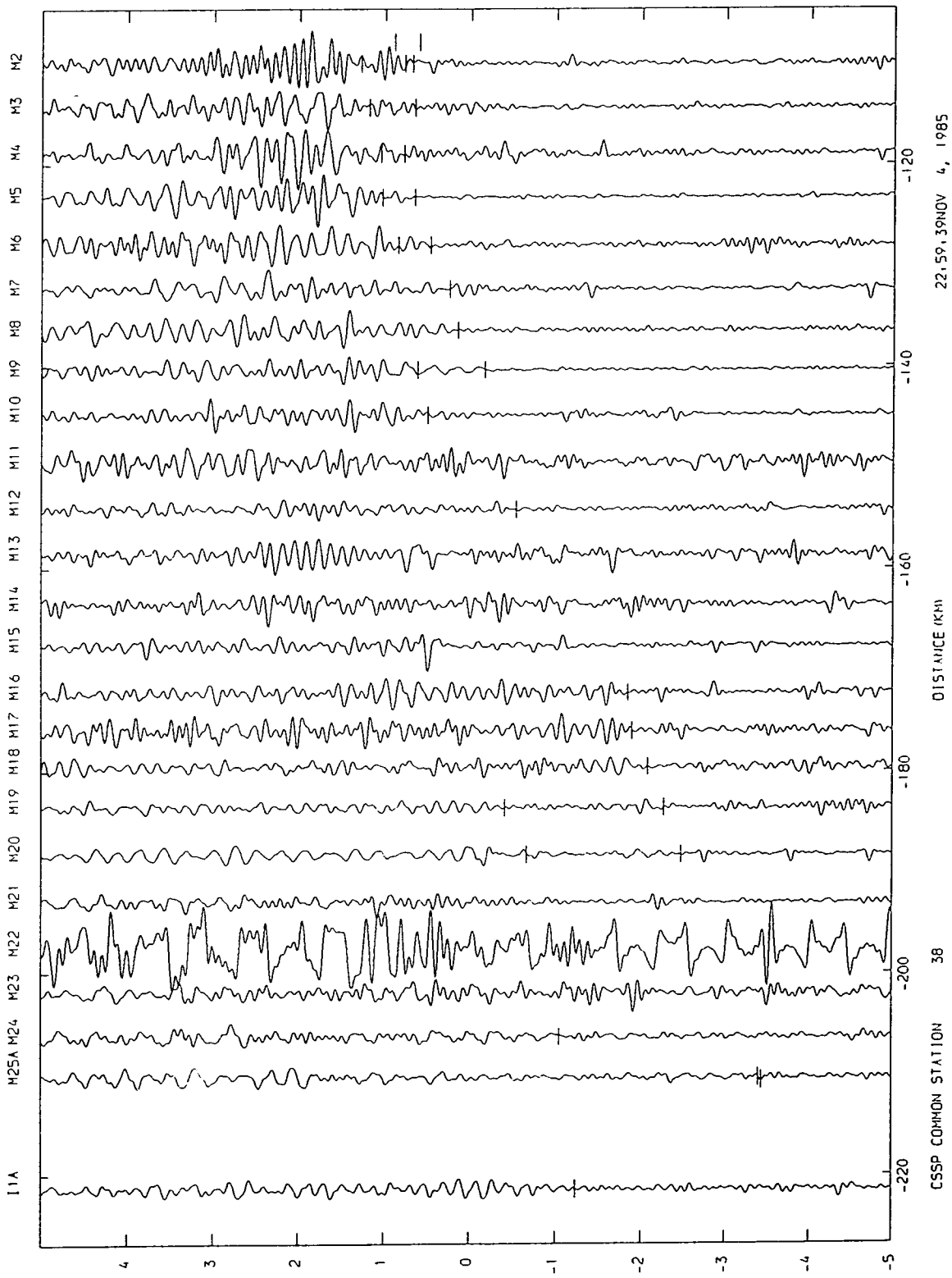
CSSP COMMON STATION 36

23:06:16NOV 4, 1985

FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0071440



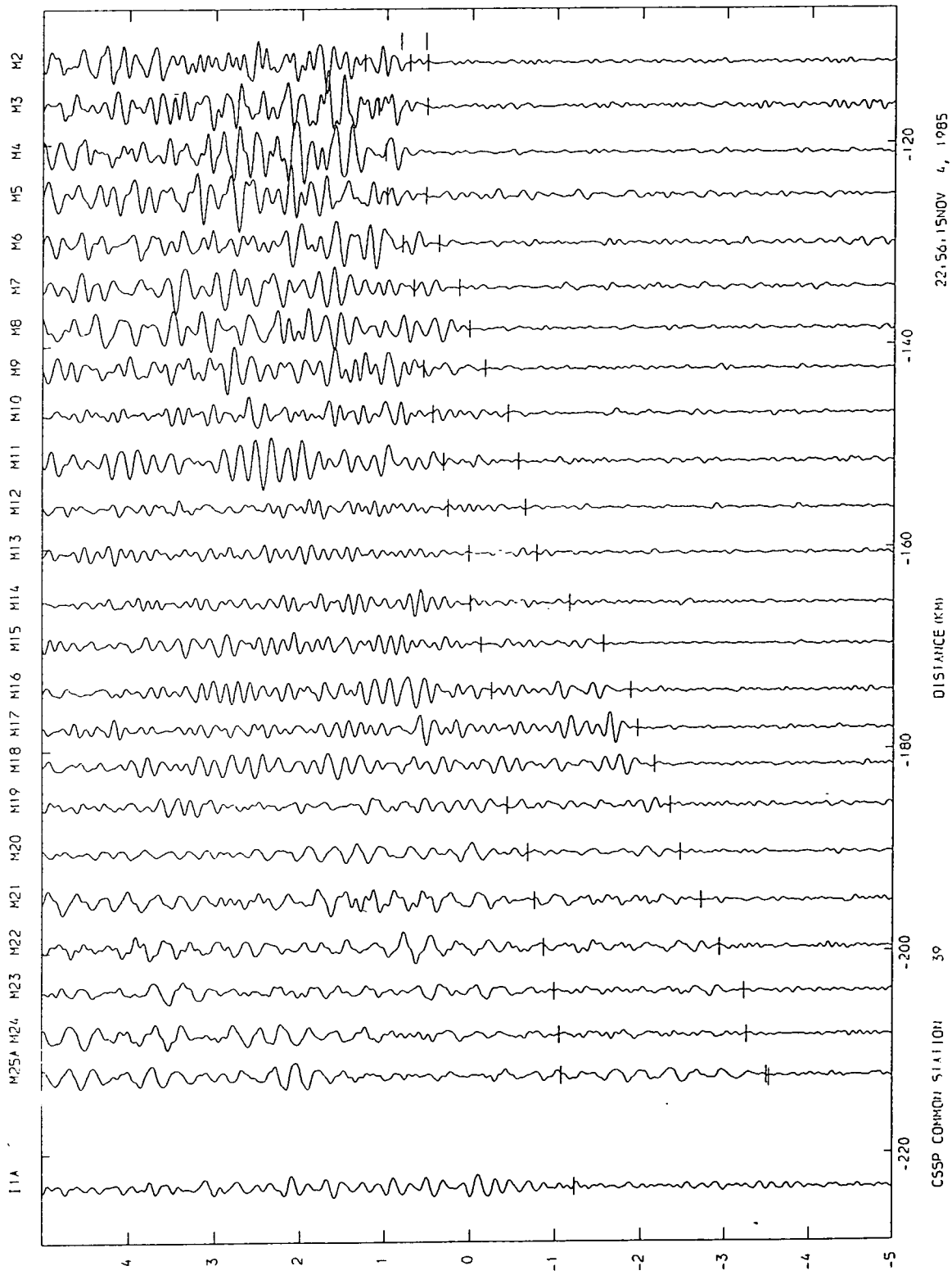
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0014603



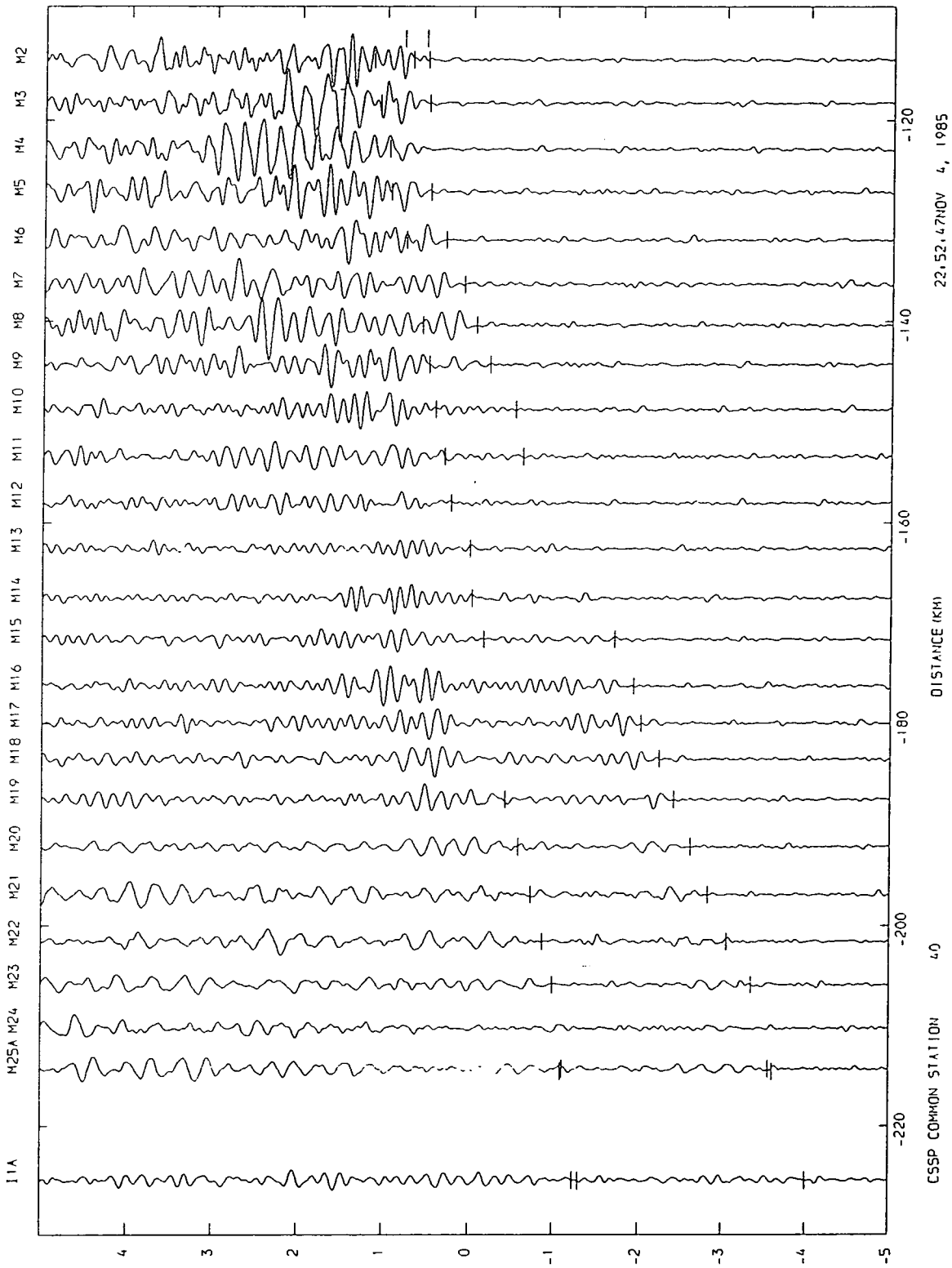
22:59:39NOV 4, 1985

CSSP COMMON STATION 38

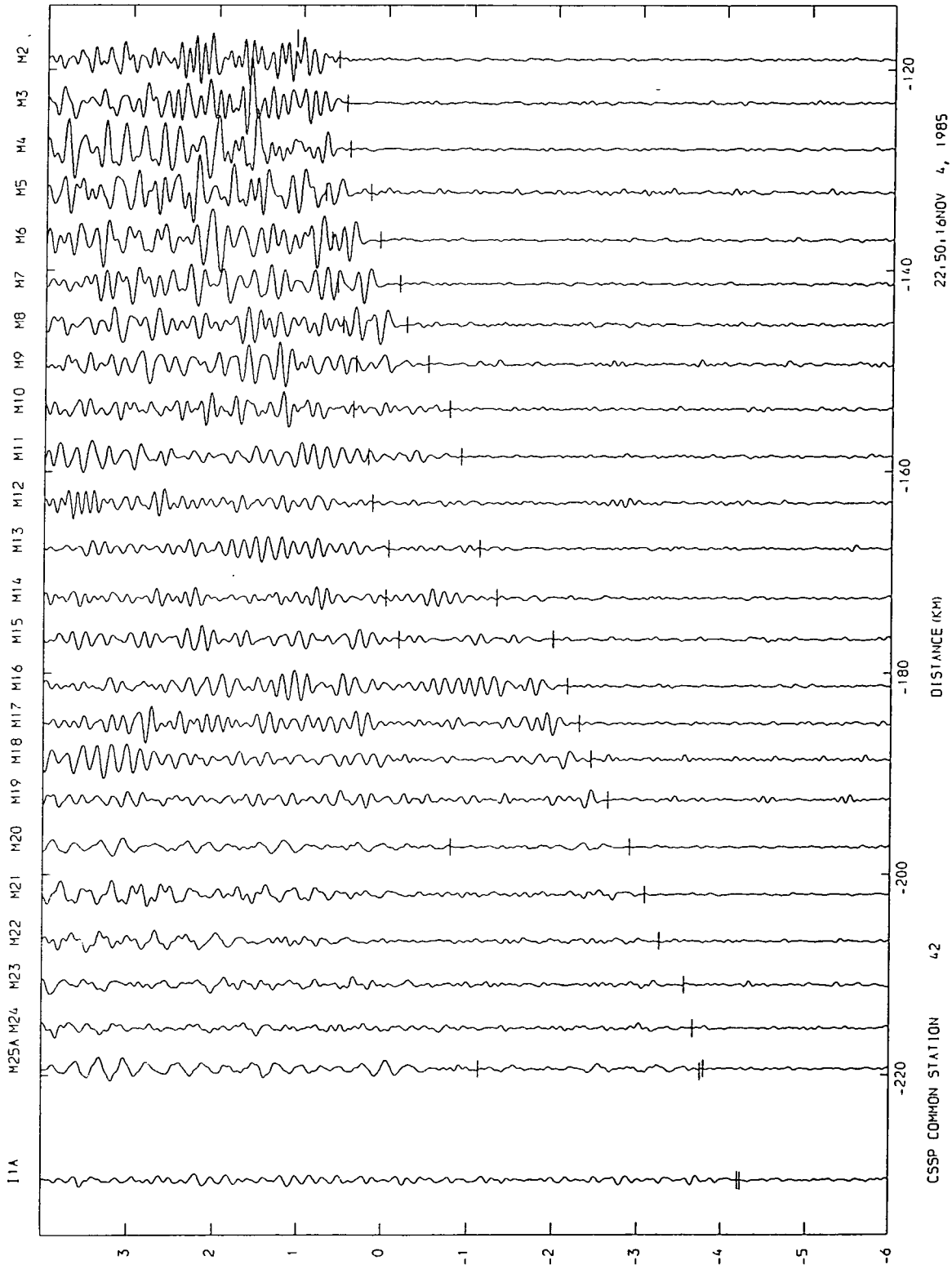
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0082327



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0108273



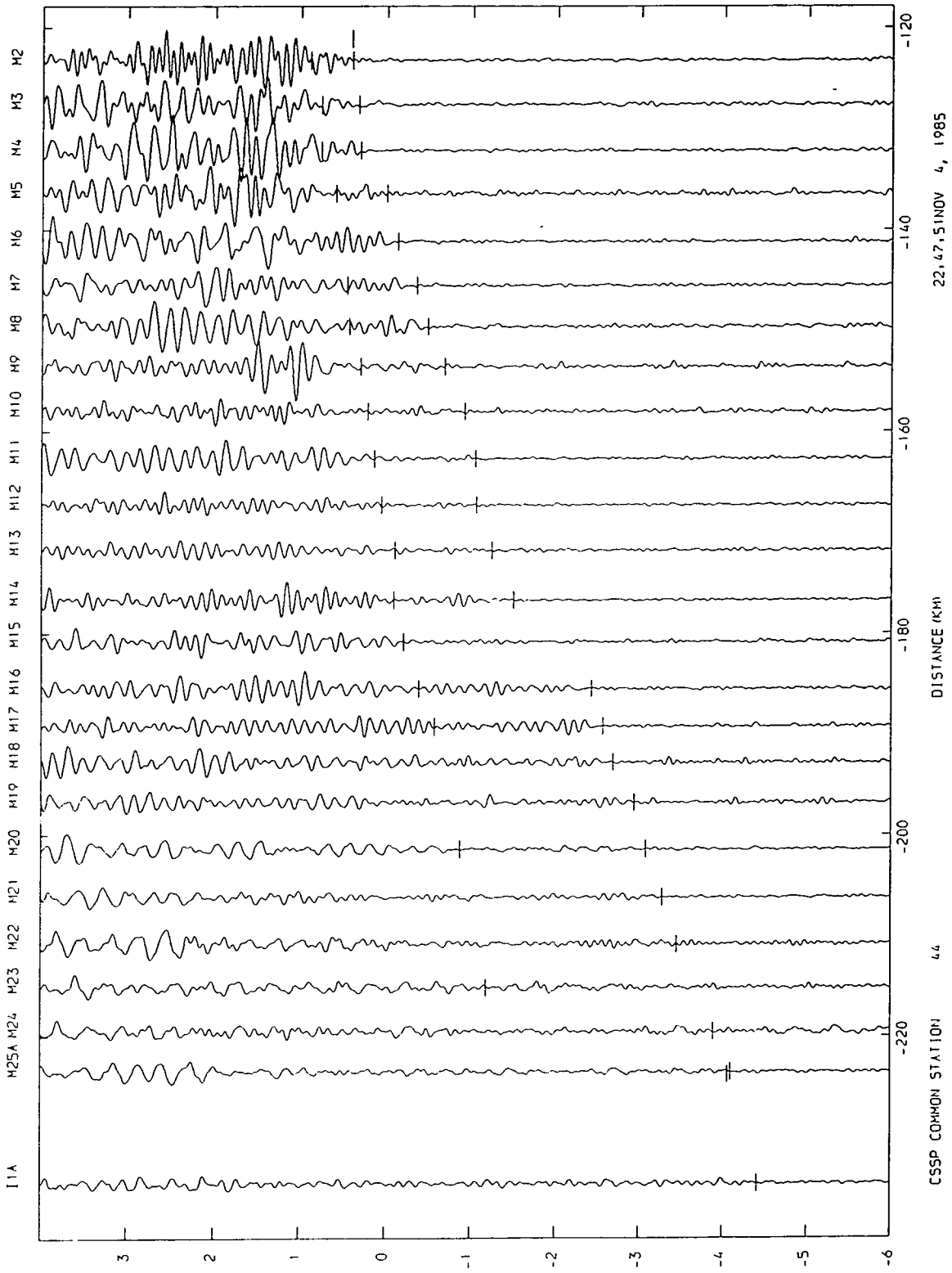
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0161674



22:50:16NOV 4, 1985

CSSP COMMON STATION 42

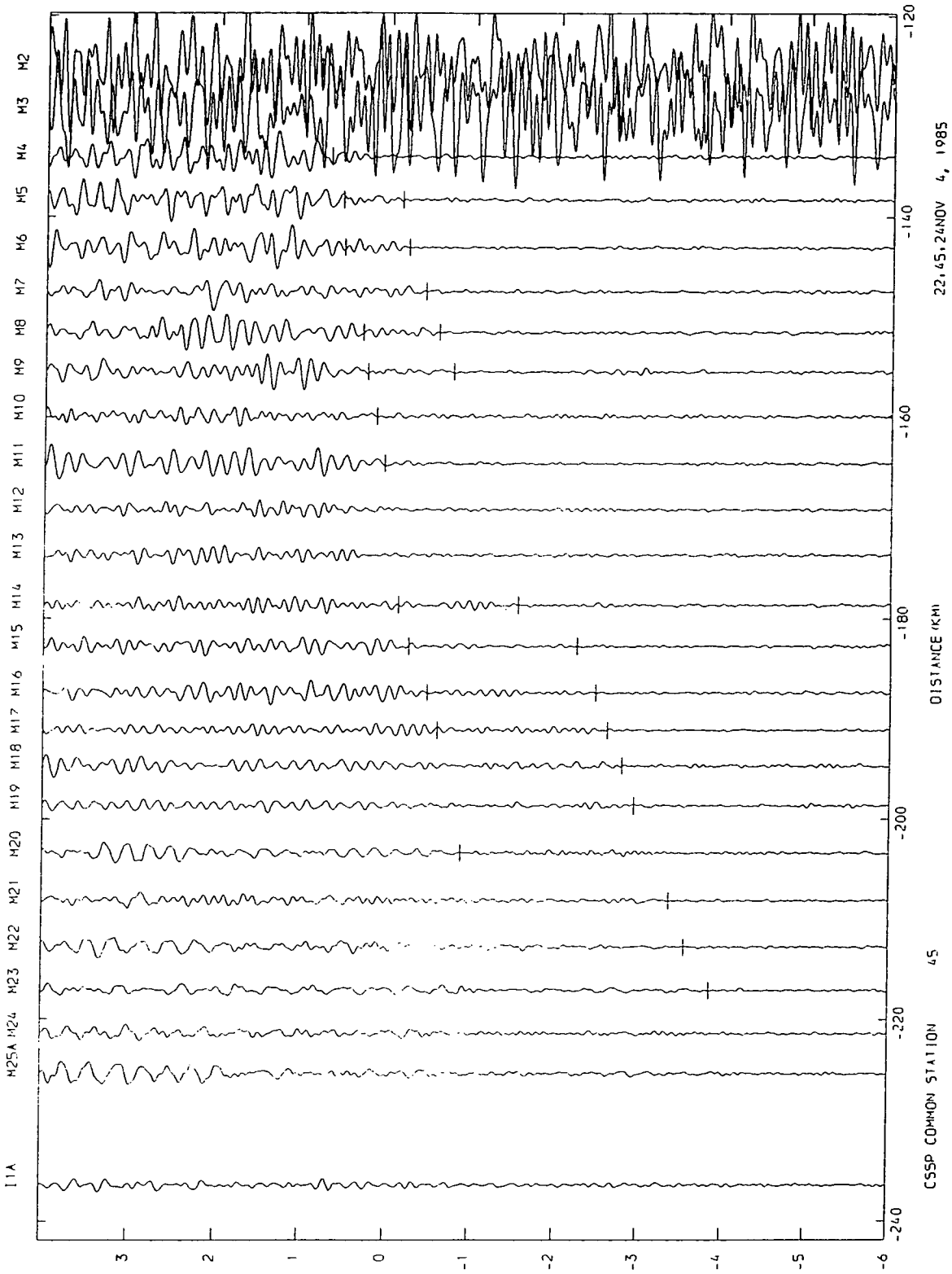
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0136858



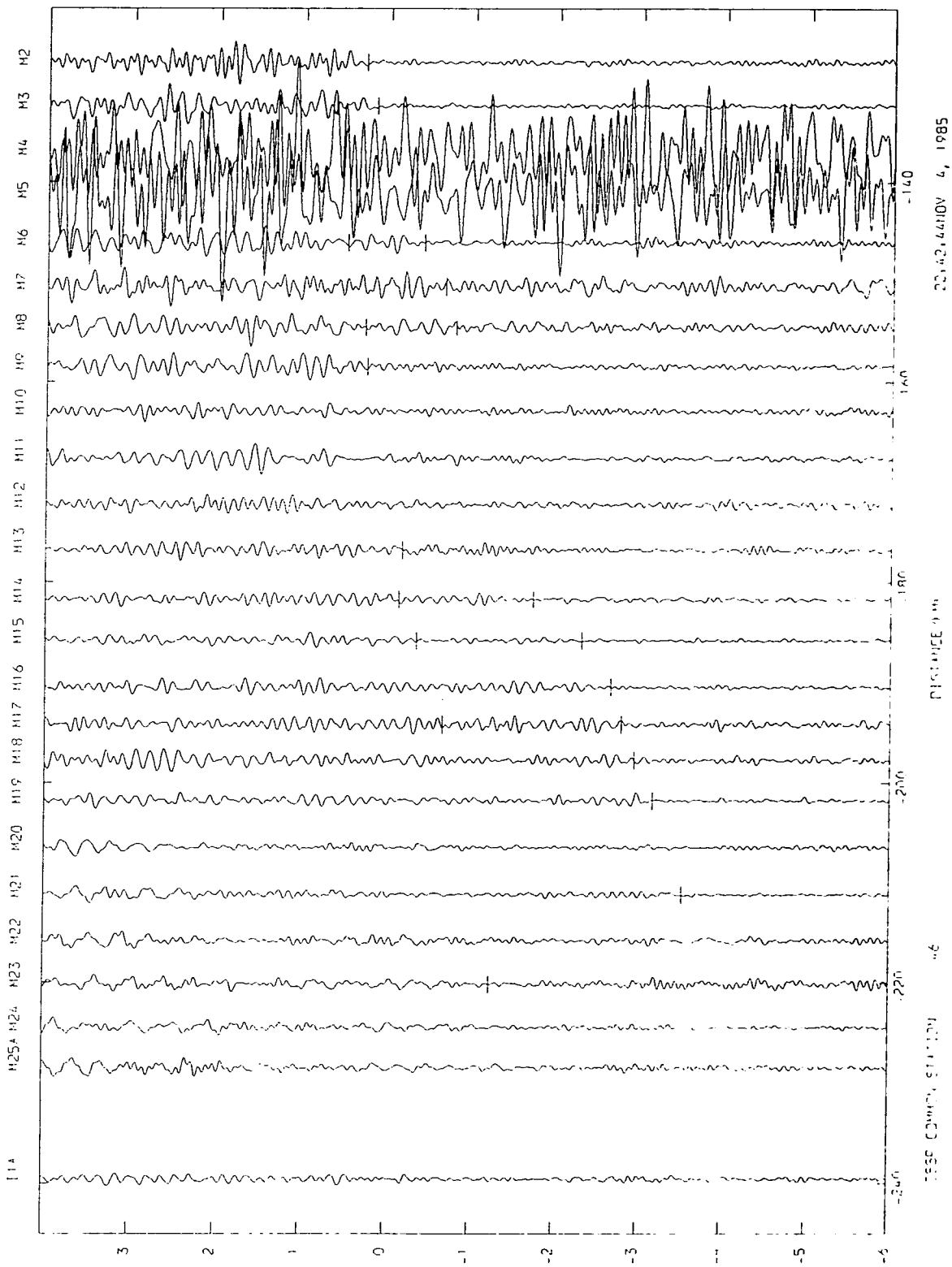
22,47,51NOV 4, 1985

CSSP COMMON STATION 44

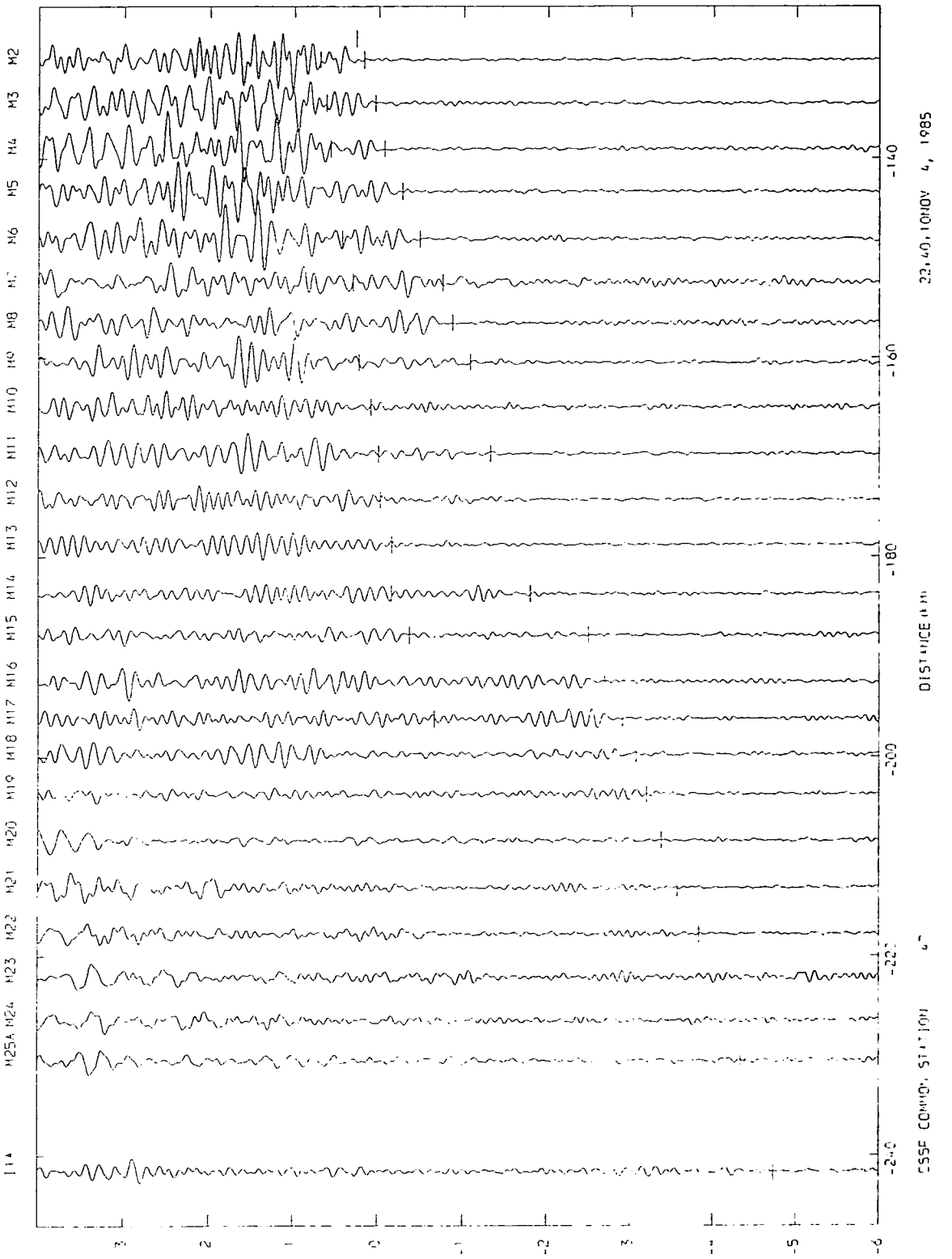
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0124930



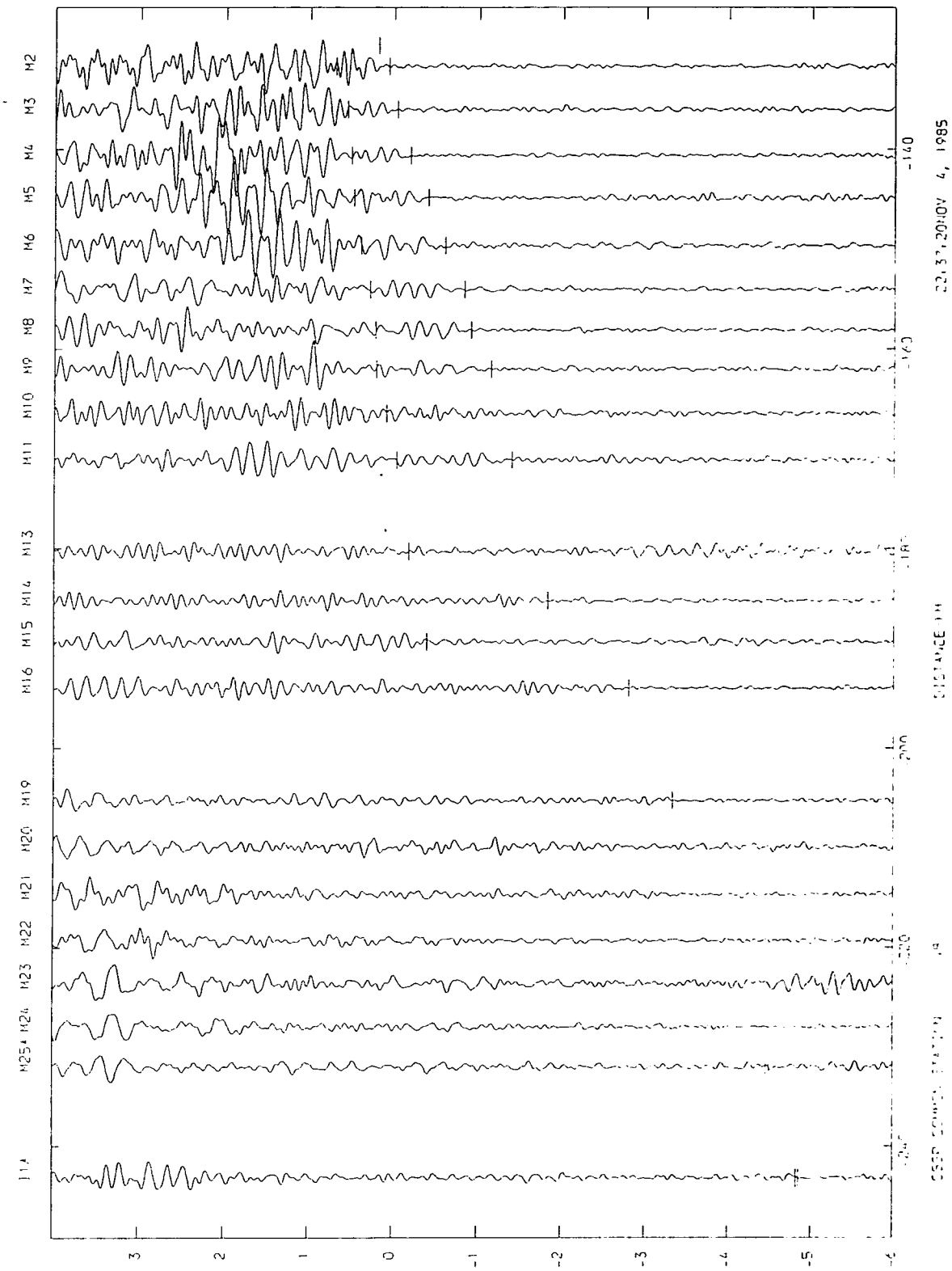
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0108243



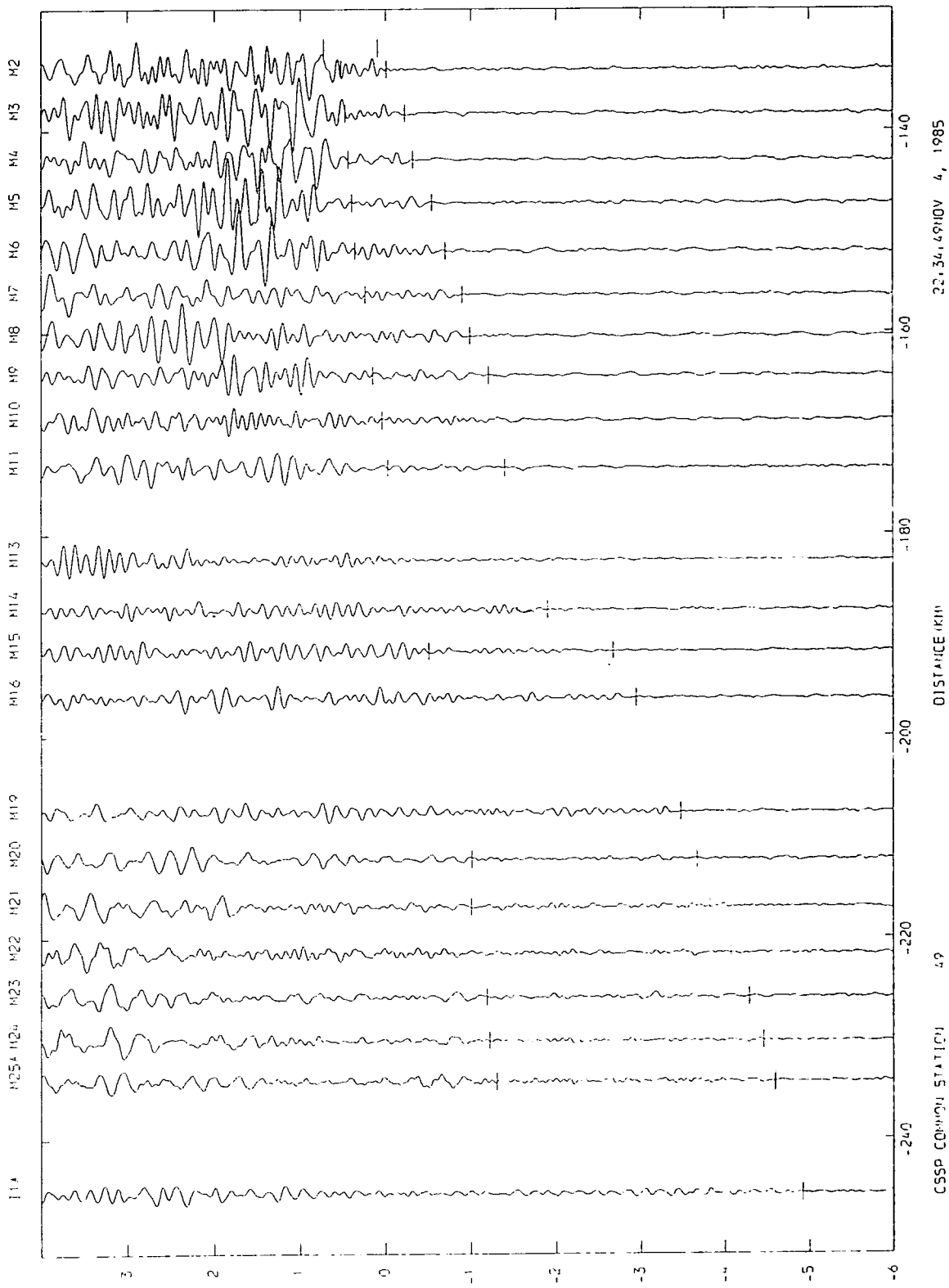
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0104169



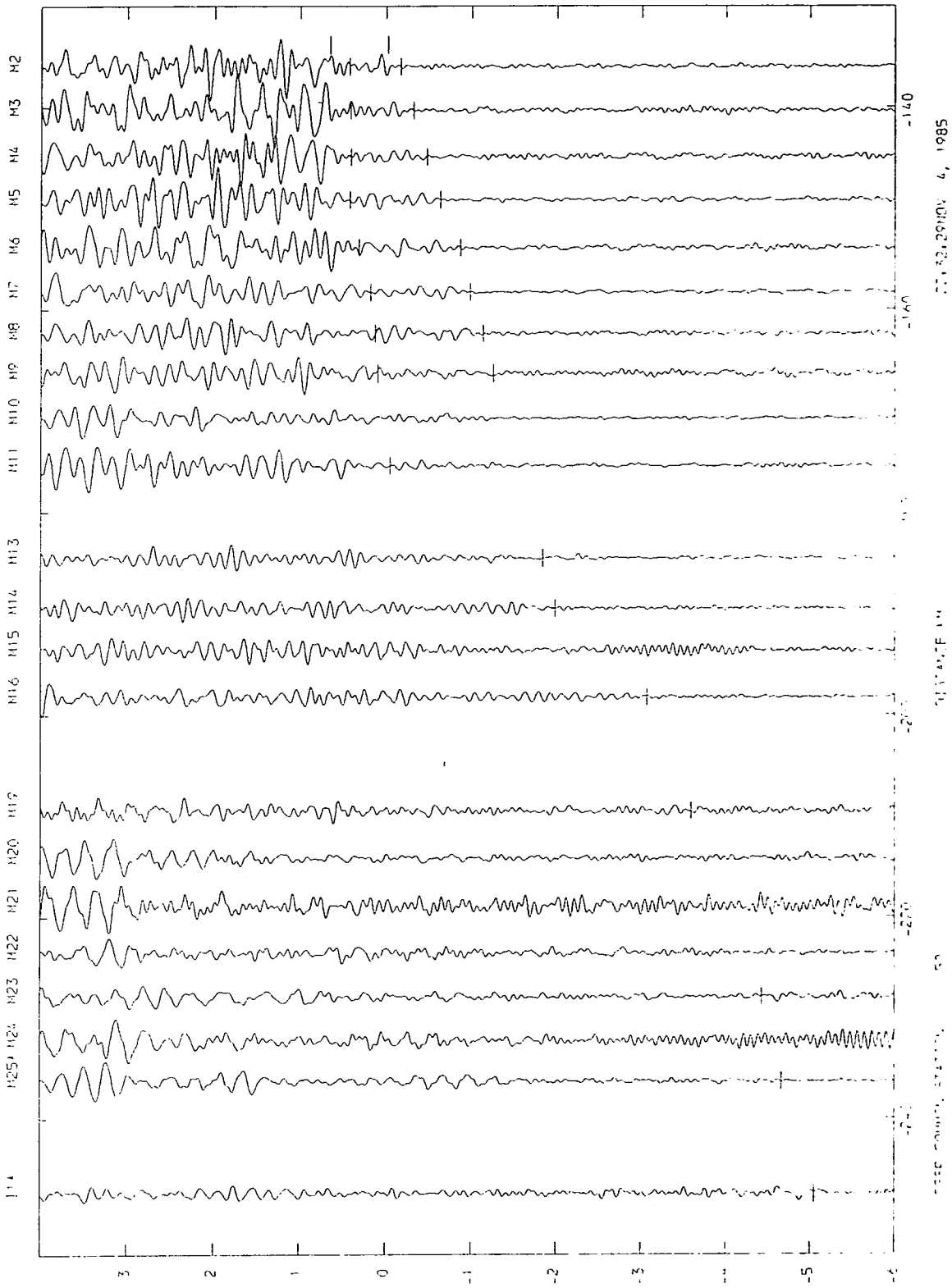
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0026935



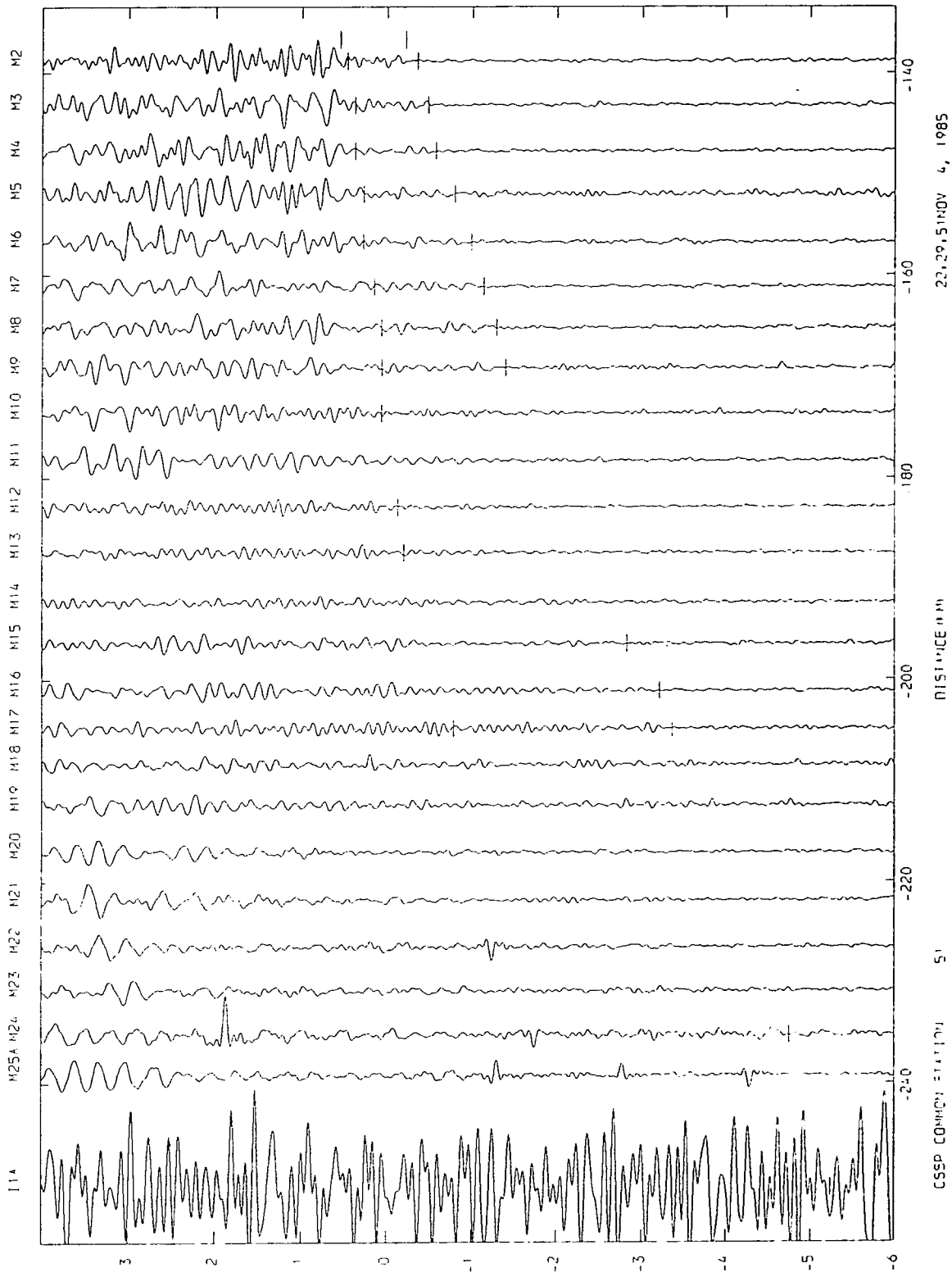
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0204456



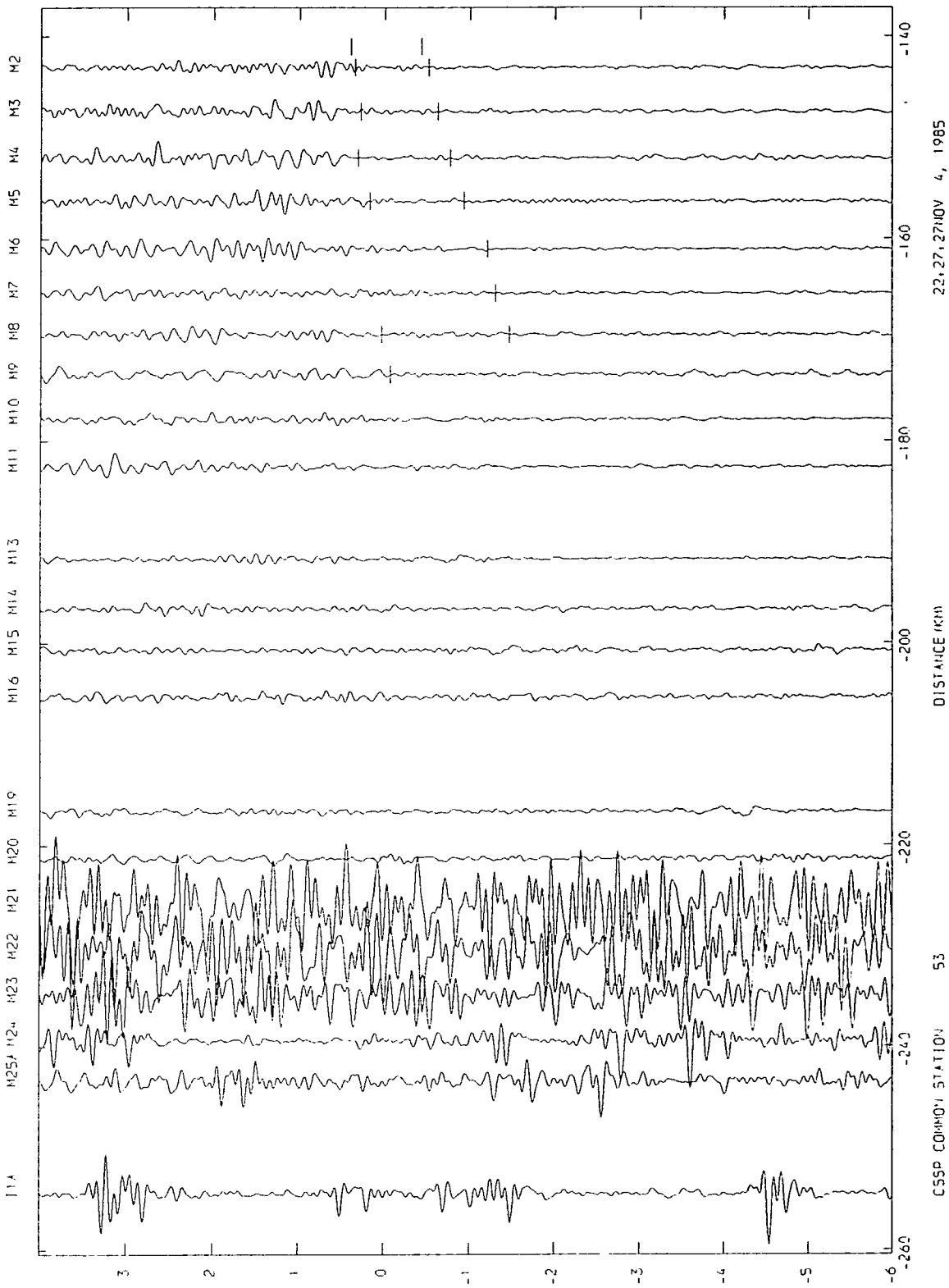
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0100433



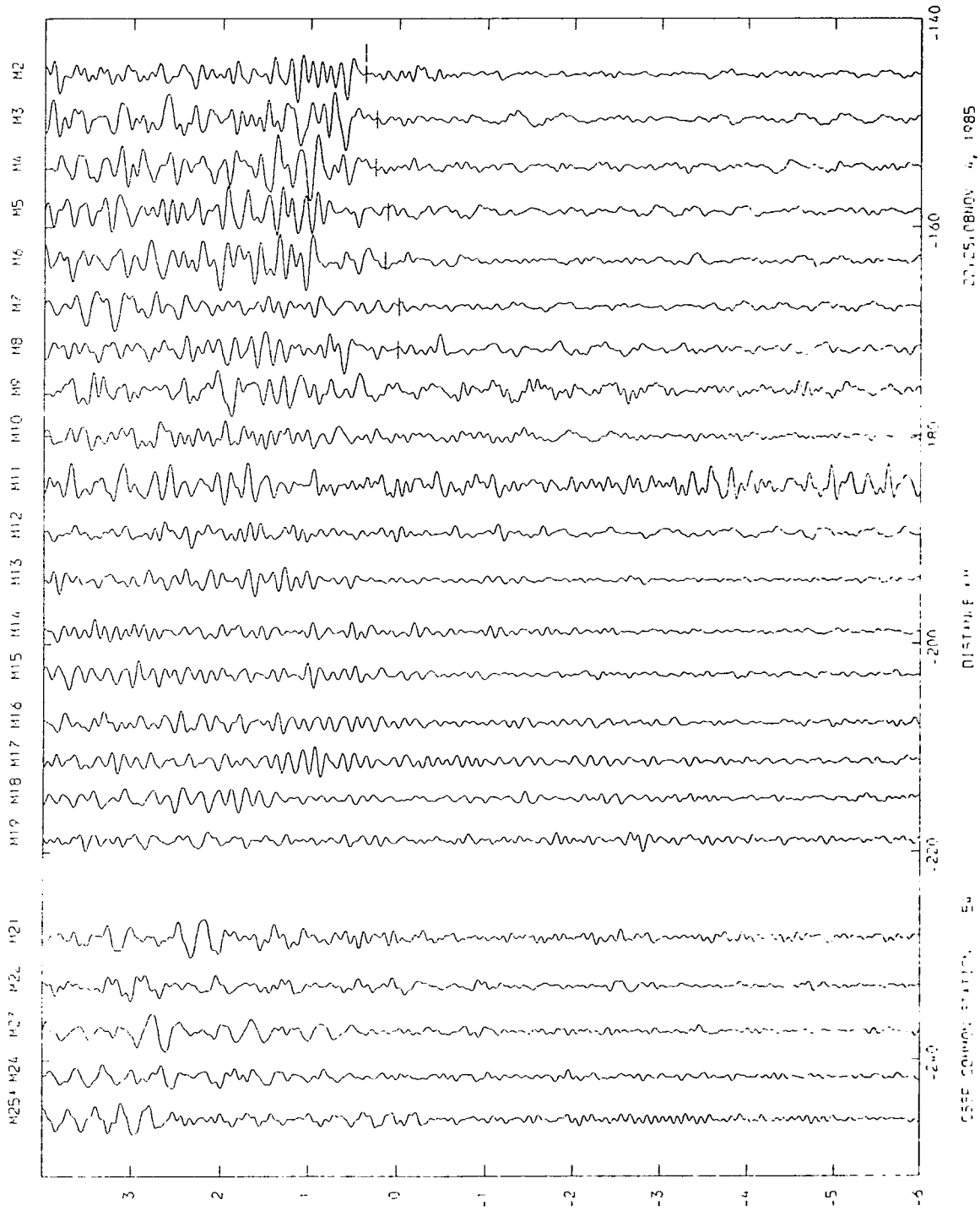
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0109266



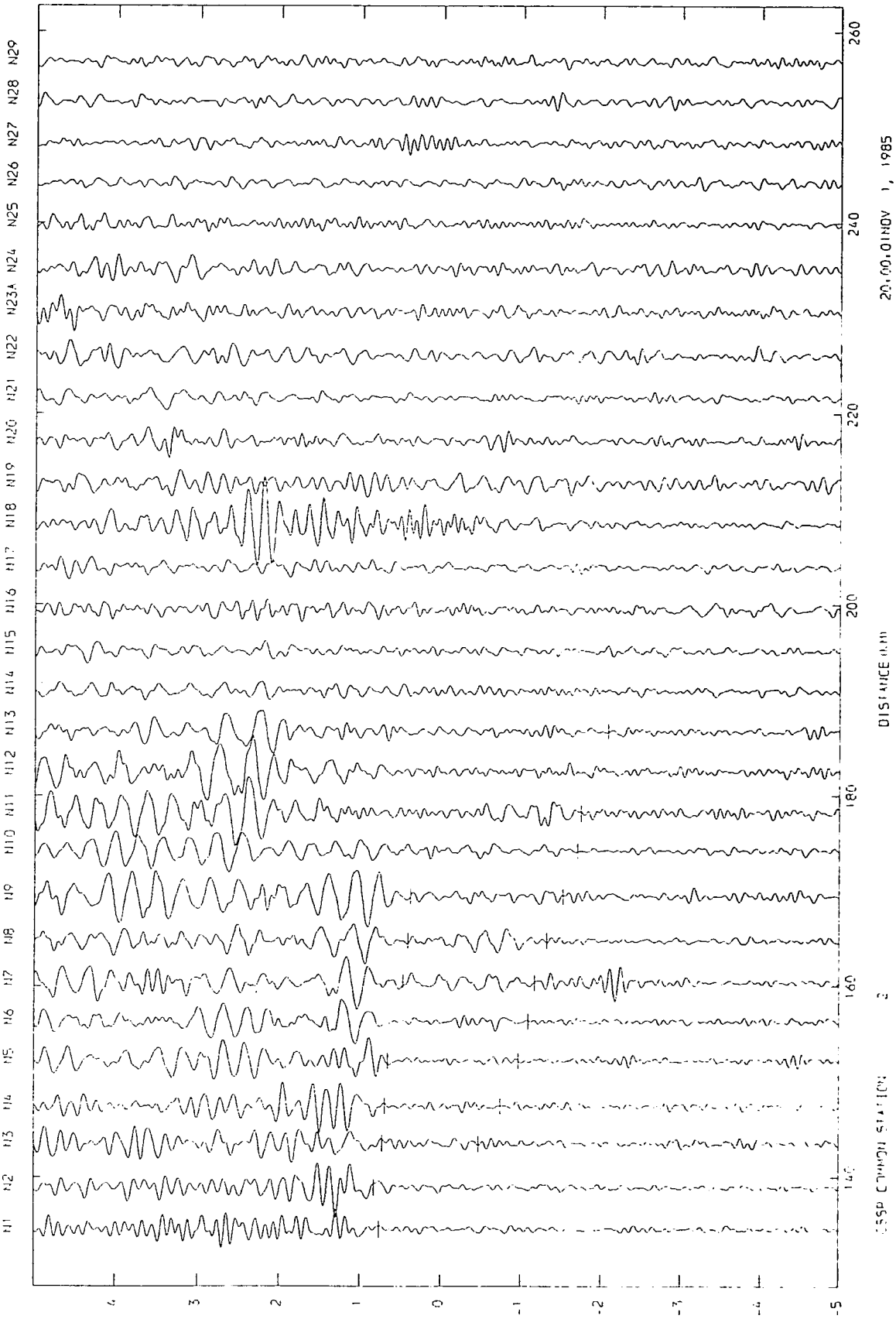
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0070107



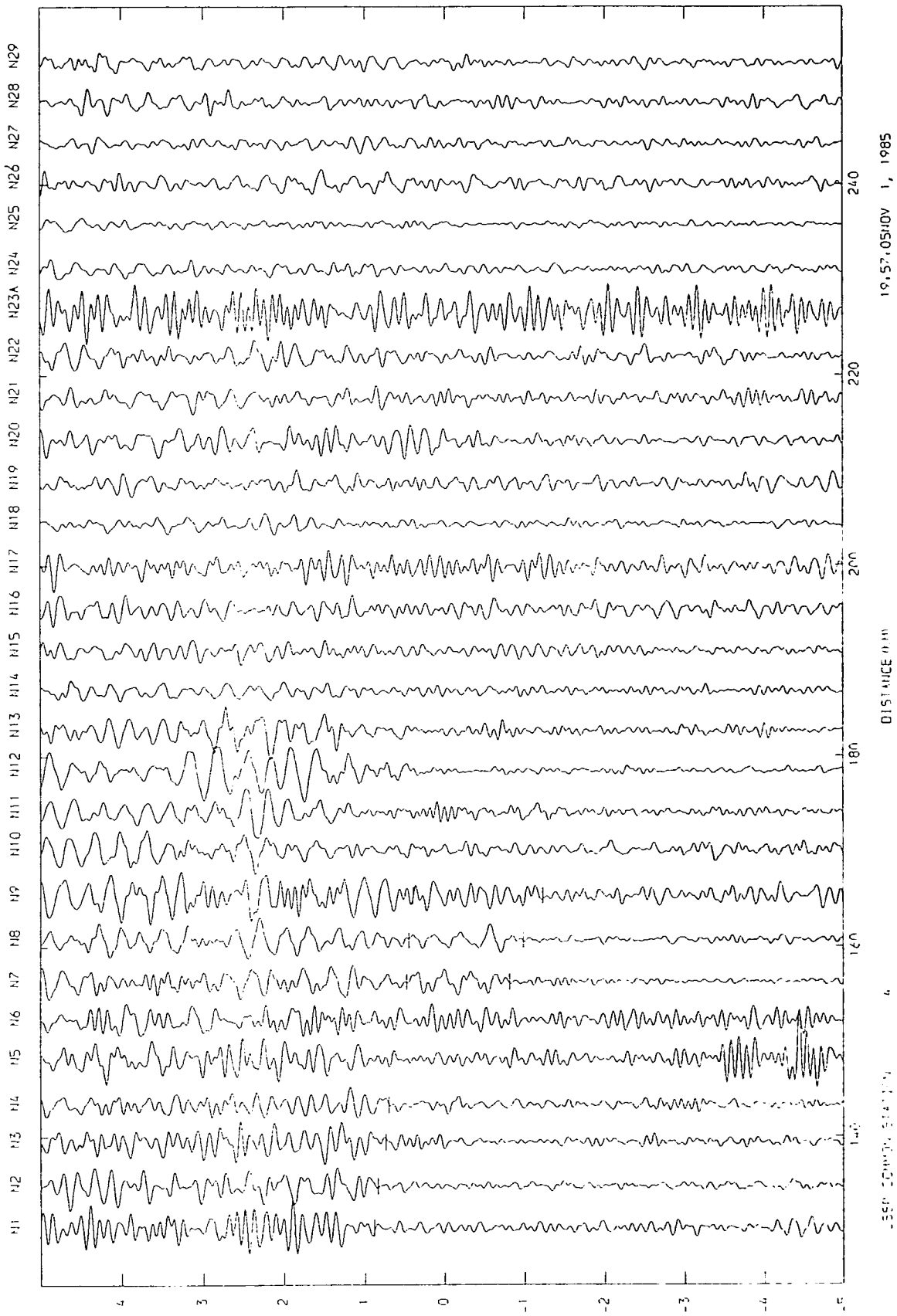
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0212678



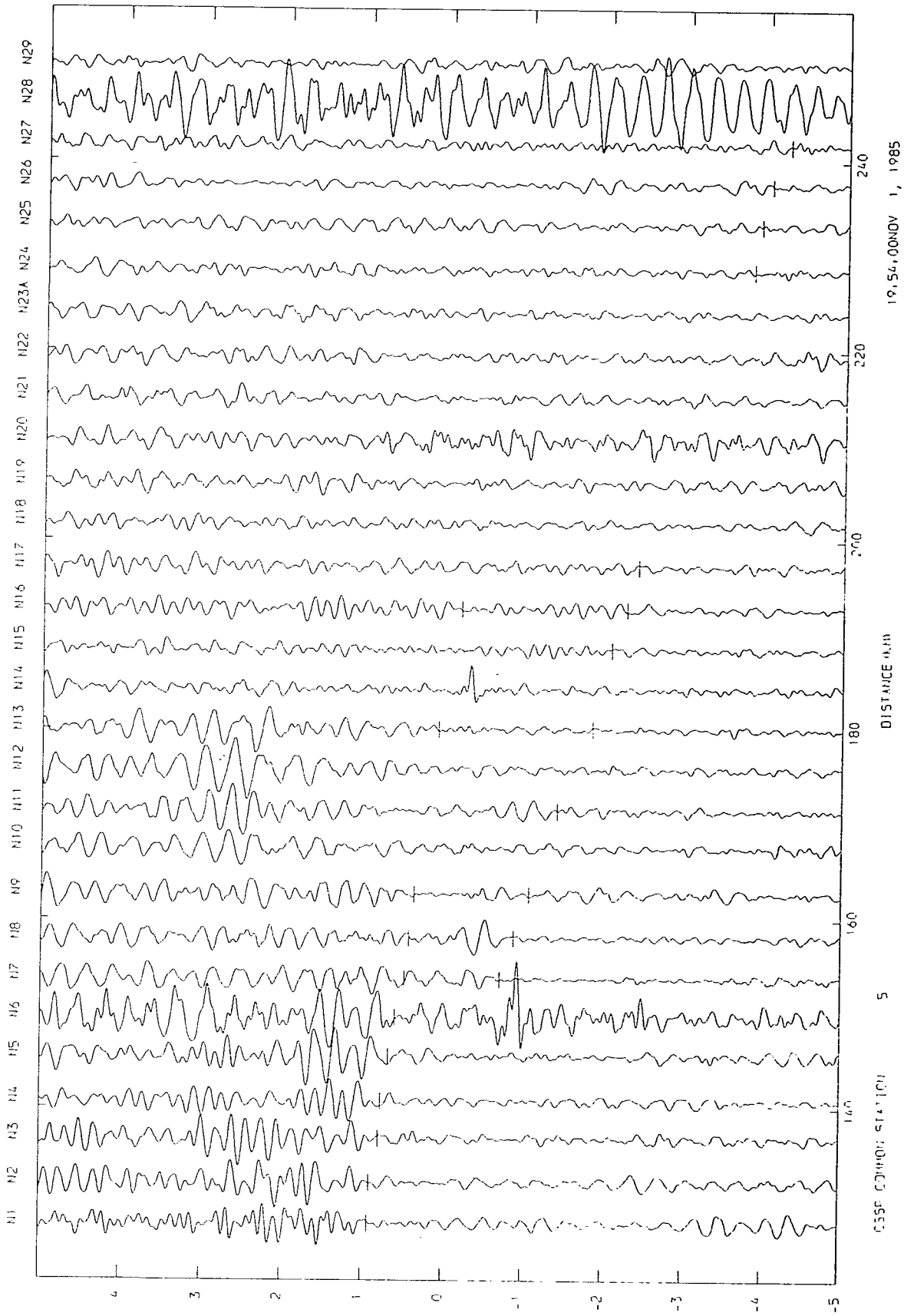
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0071838



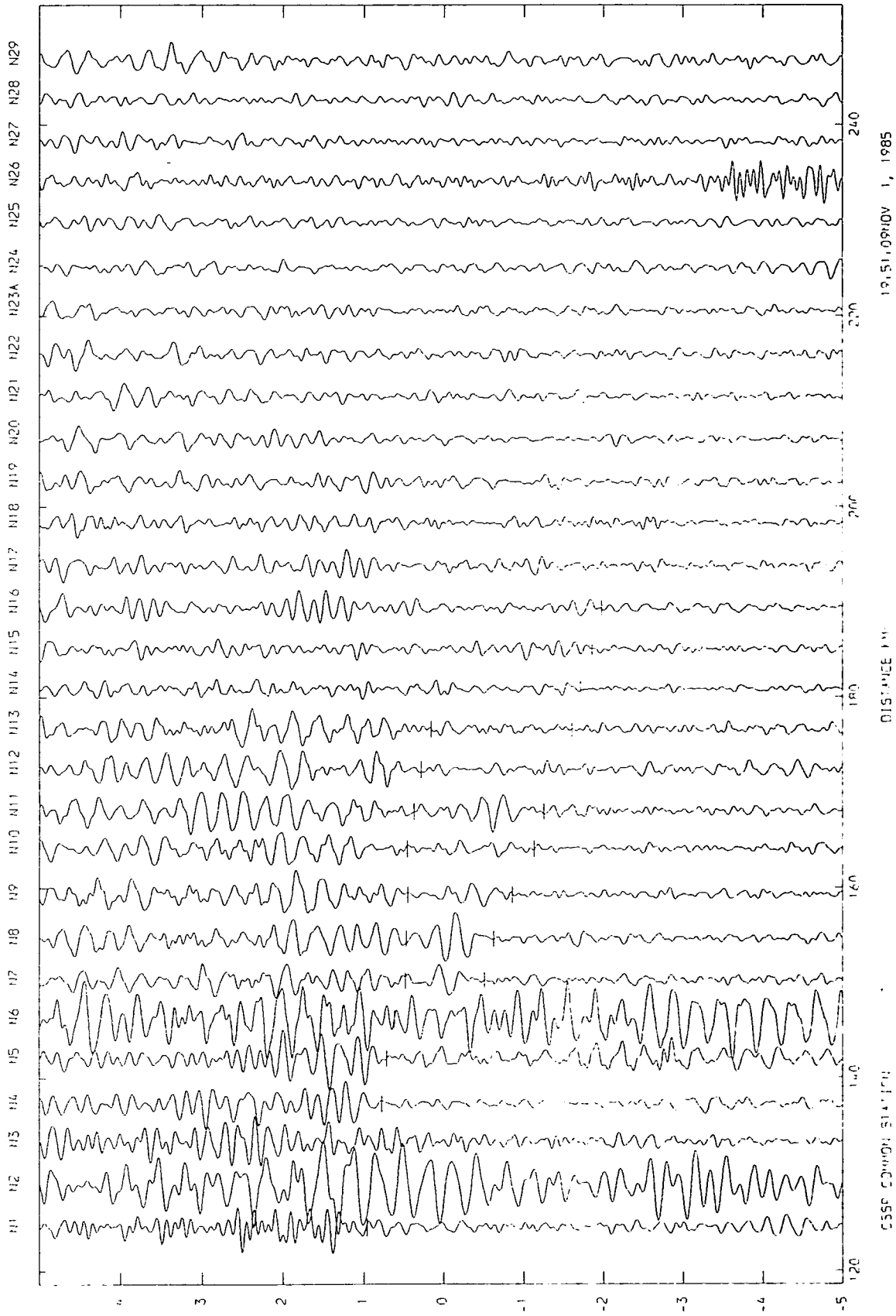
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0015149



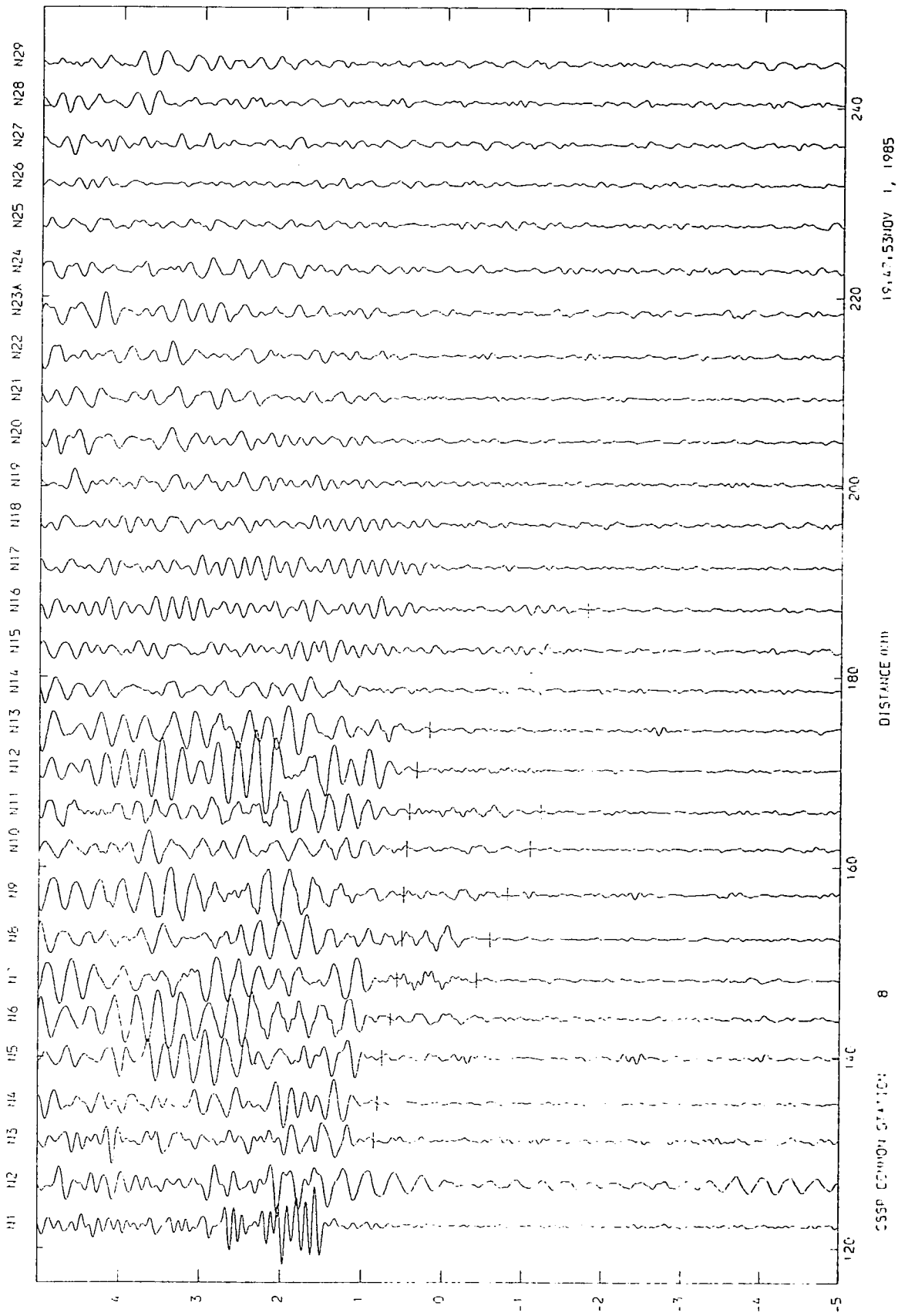
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0057881



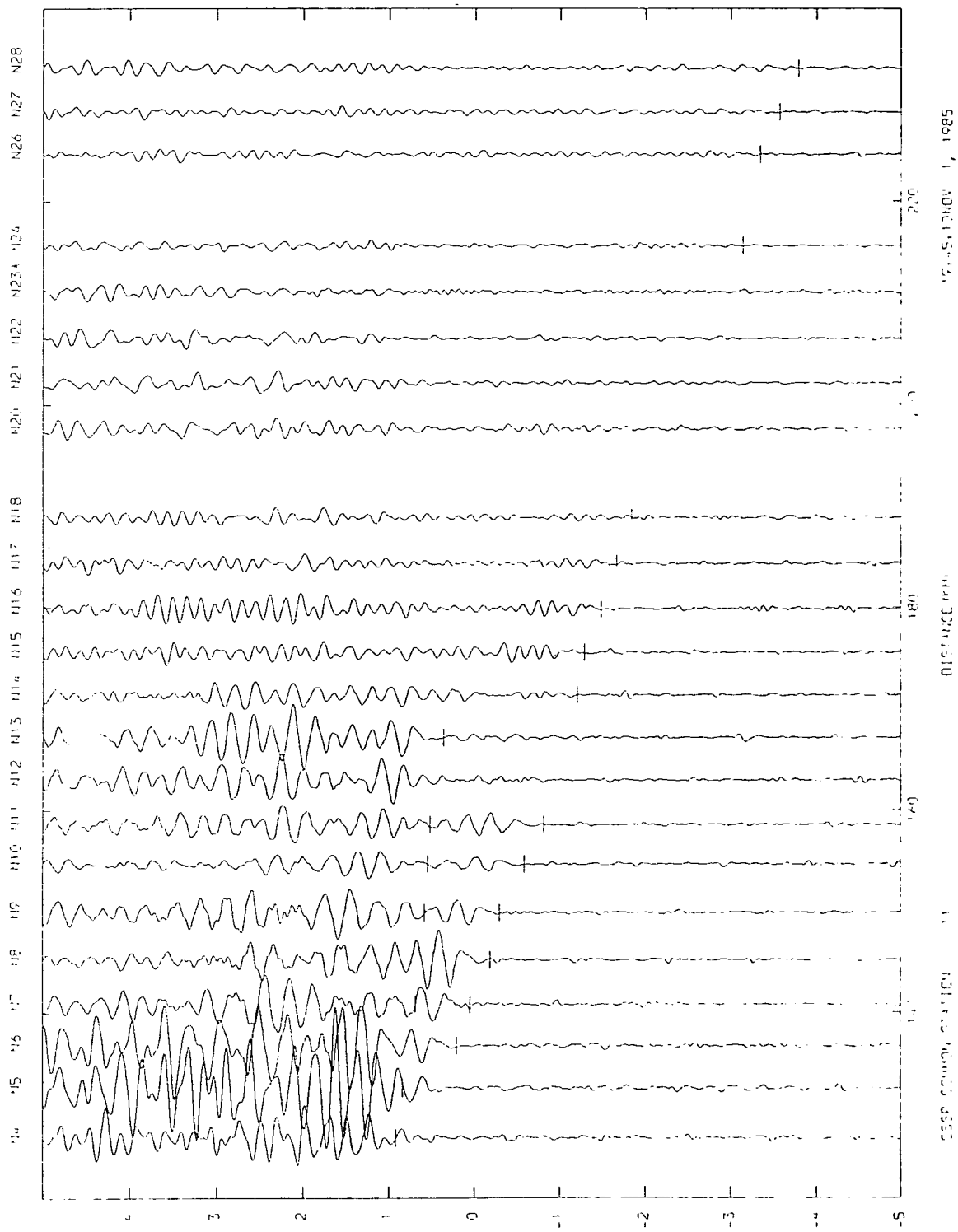
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0095995



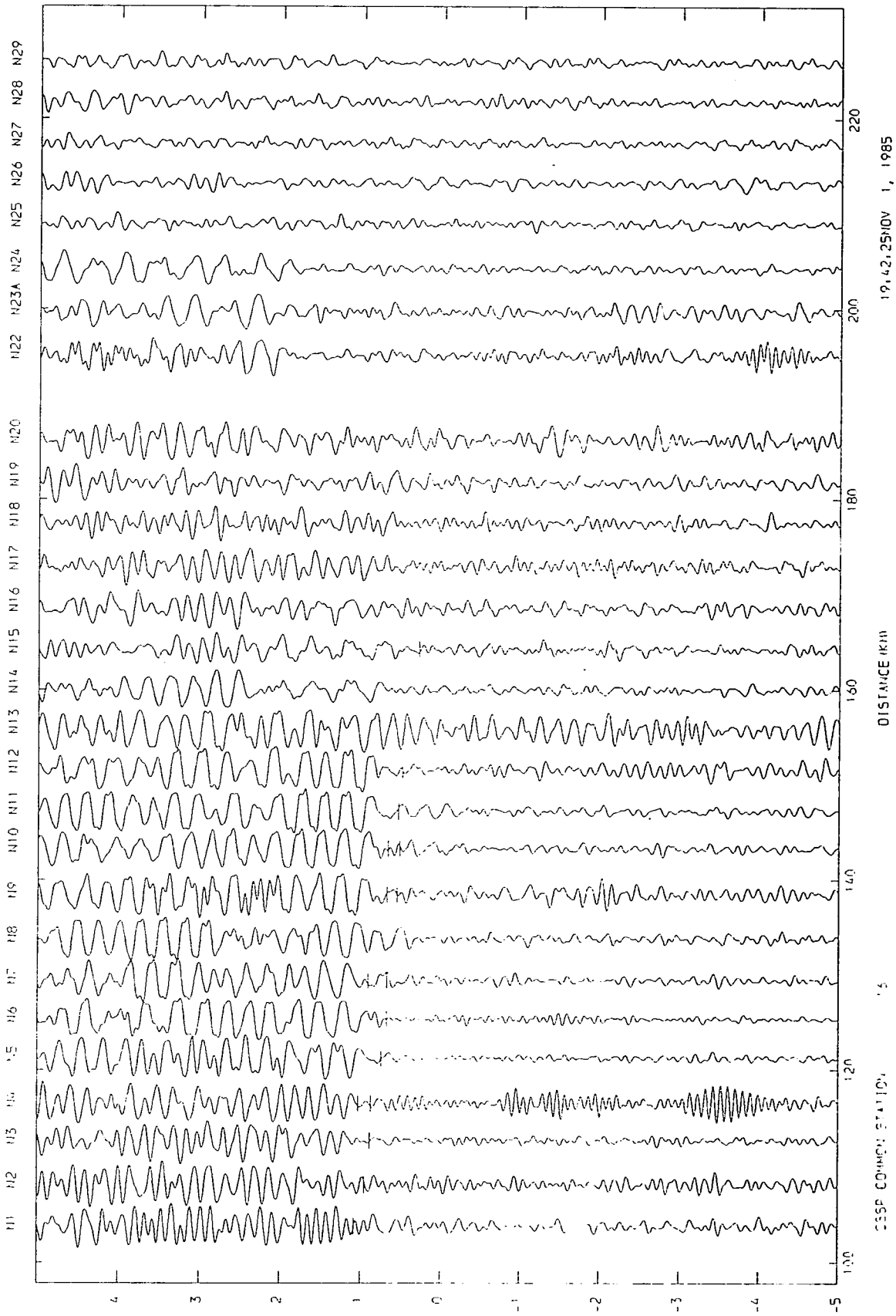
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0093000



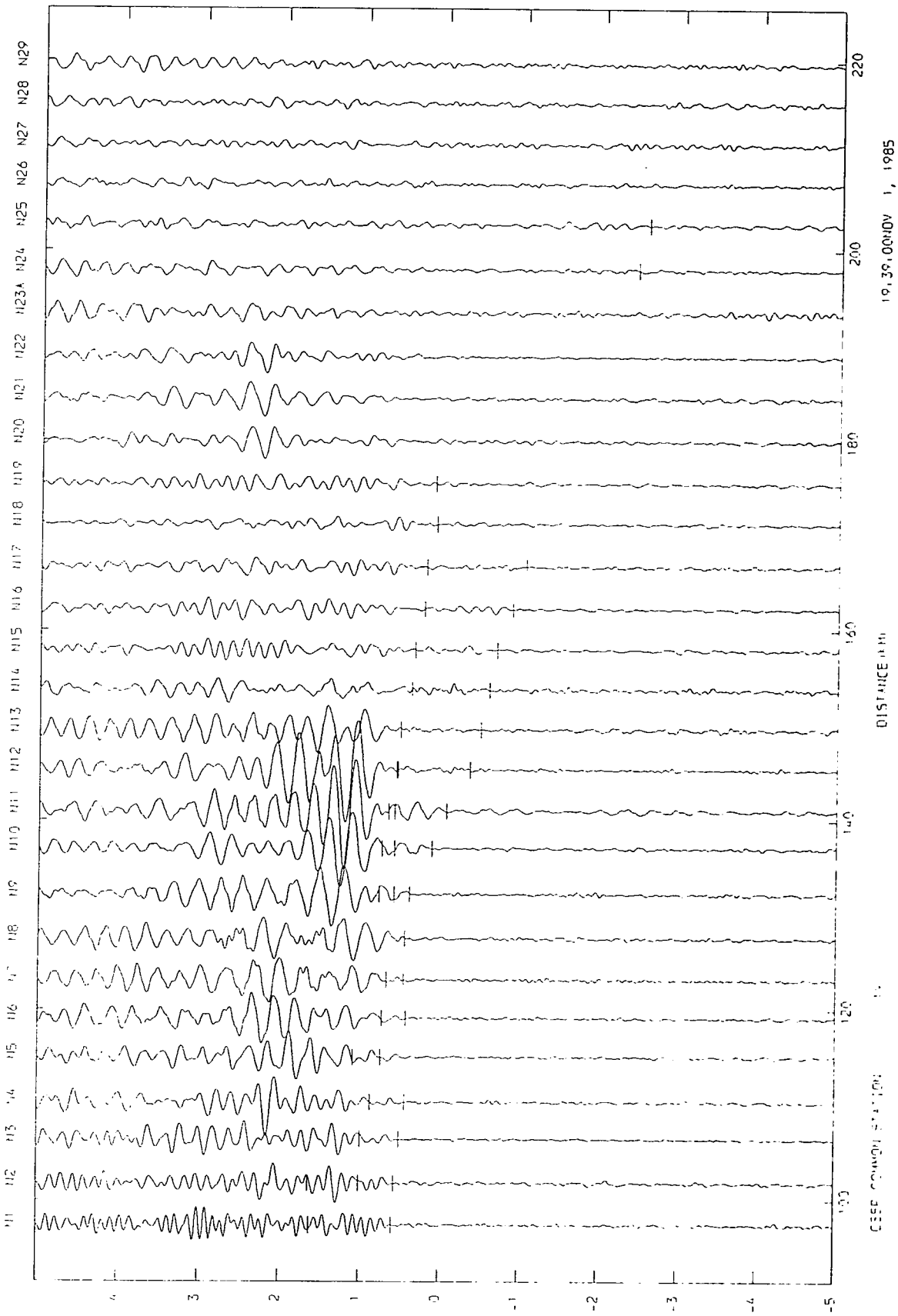
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0000049



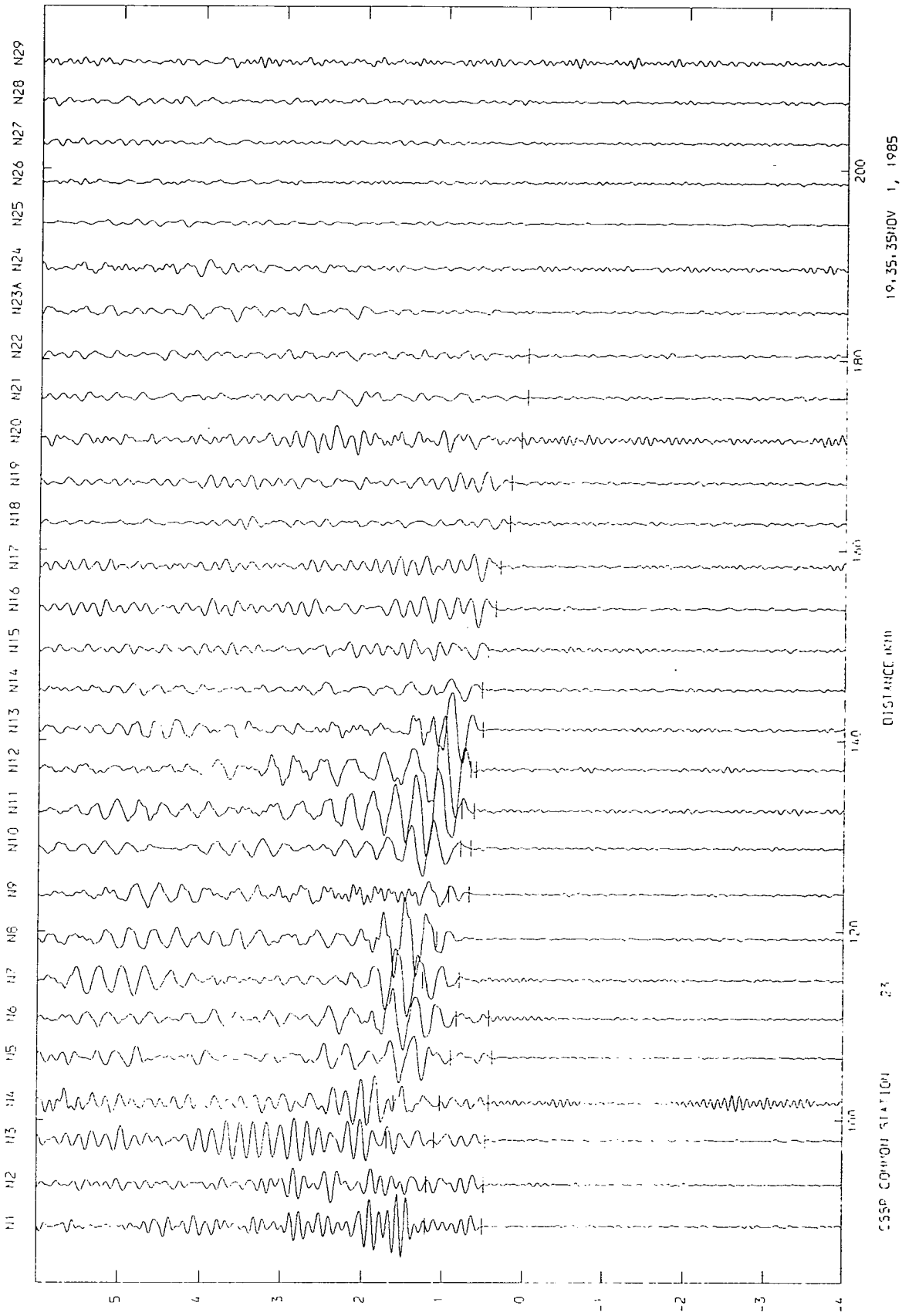
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0283044



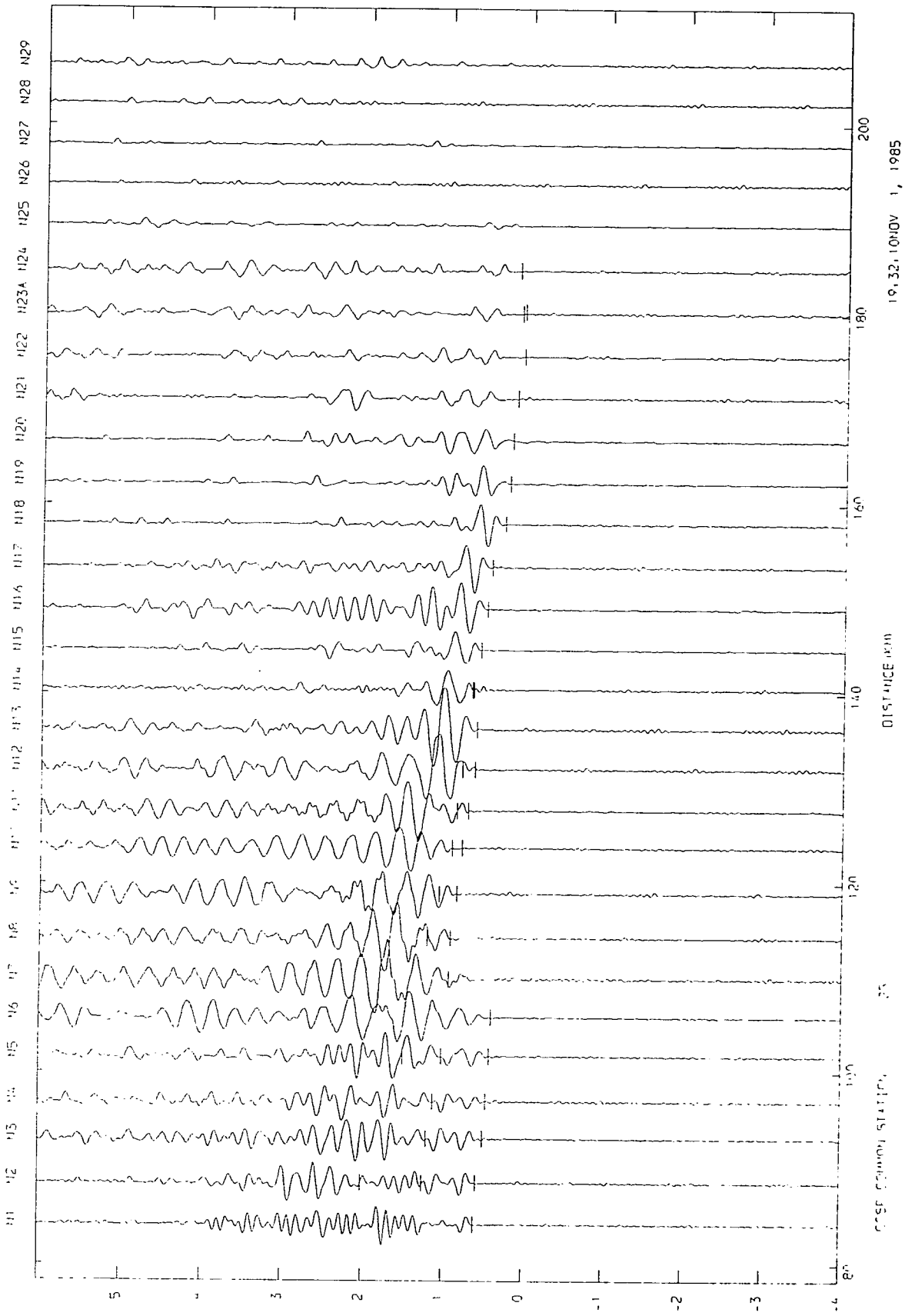
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0053170



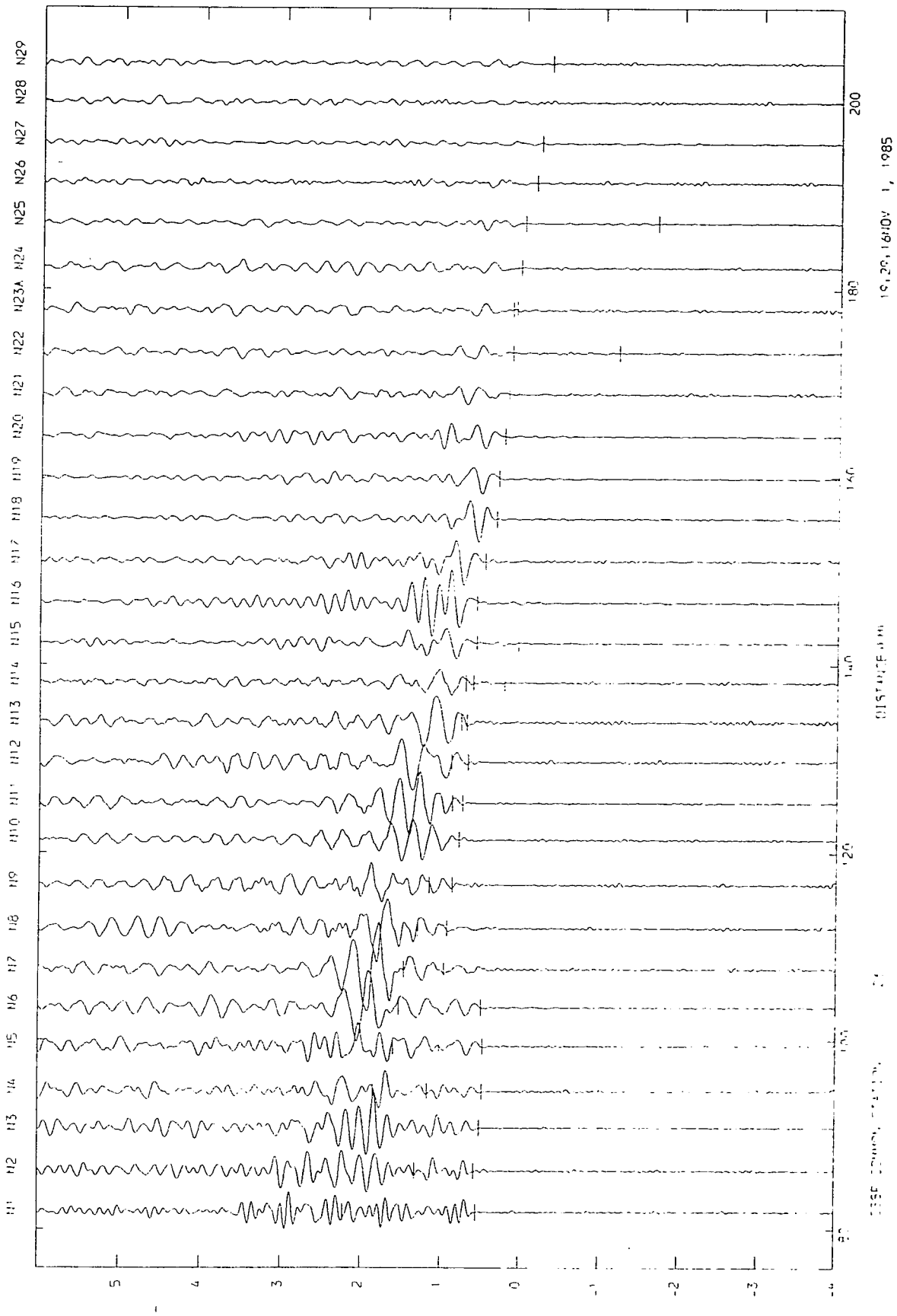
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0032165



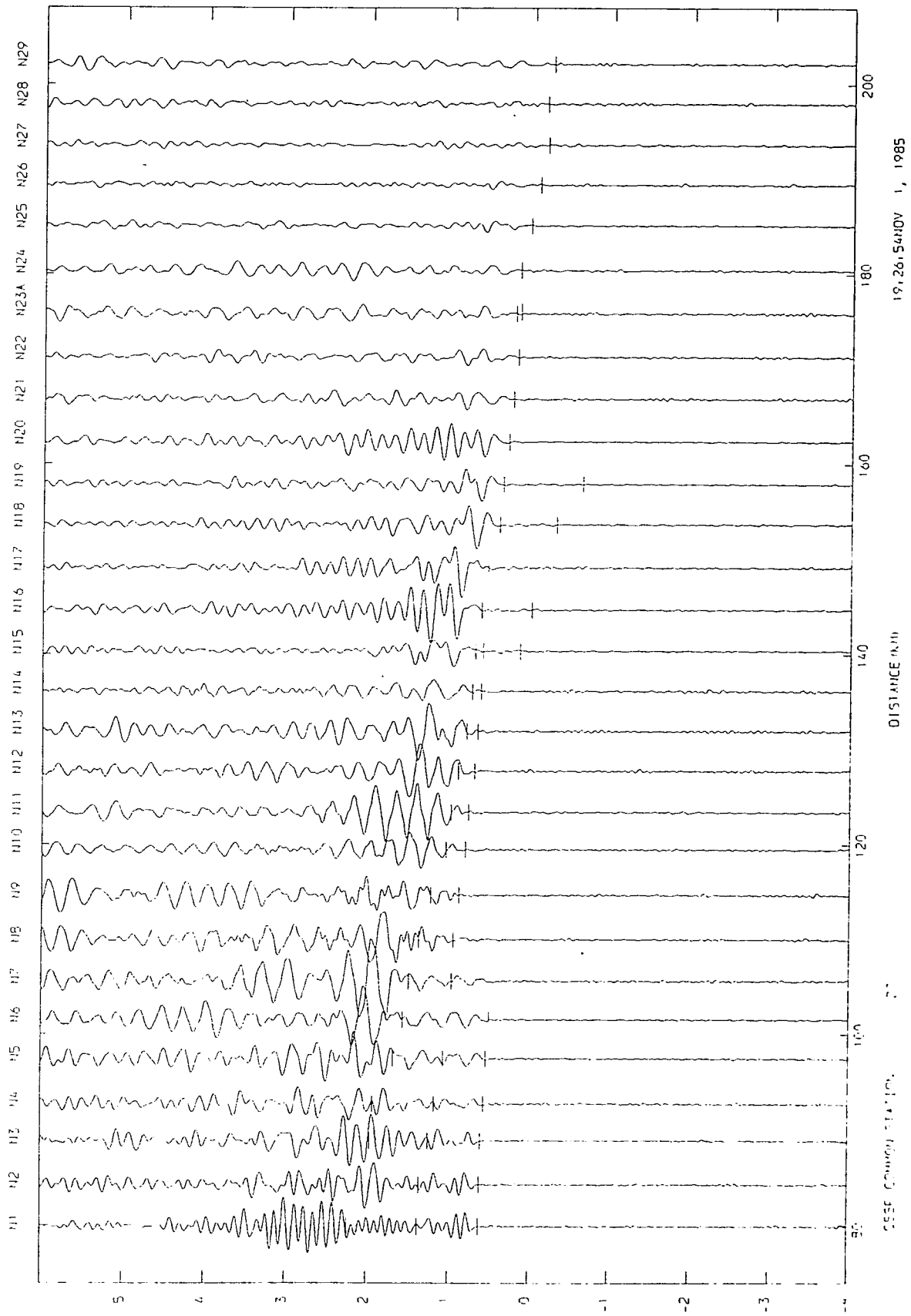
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0032970



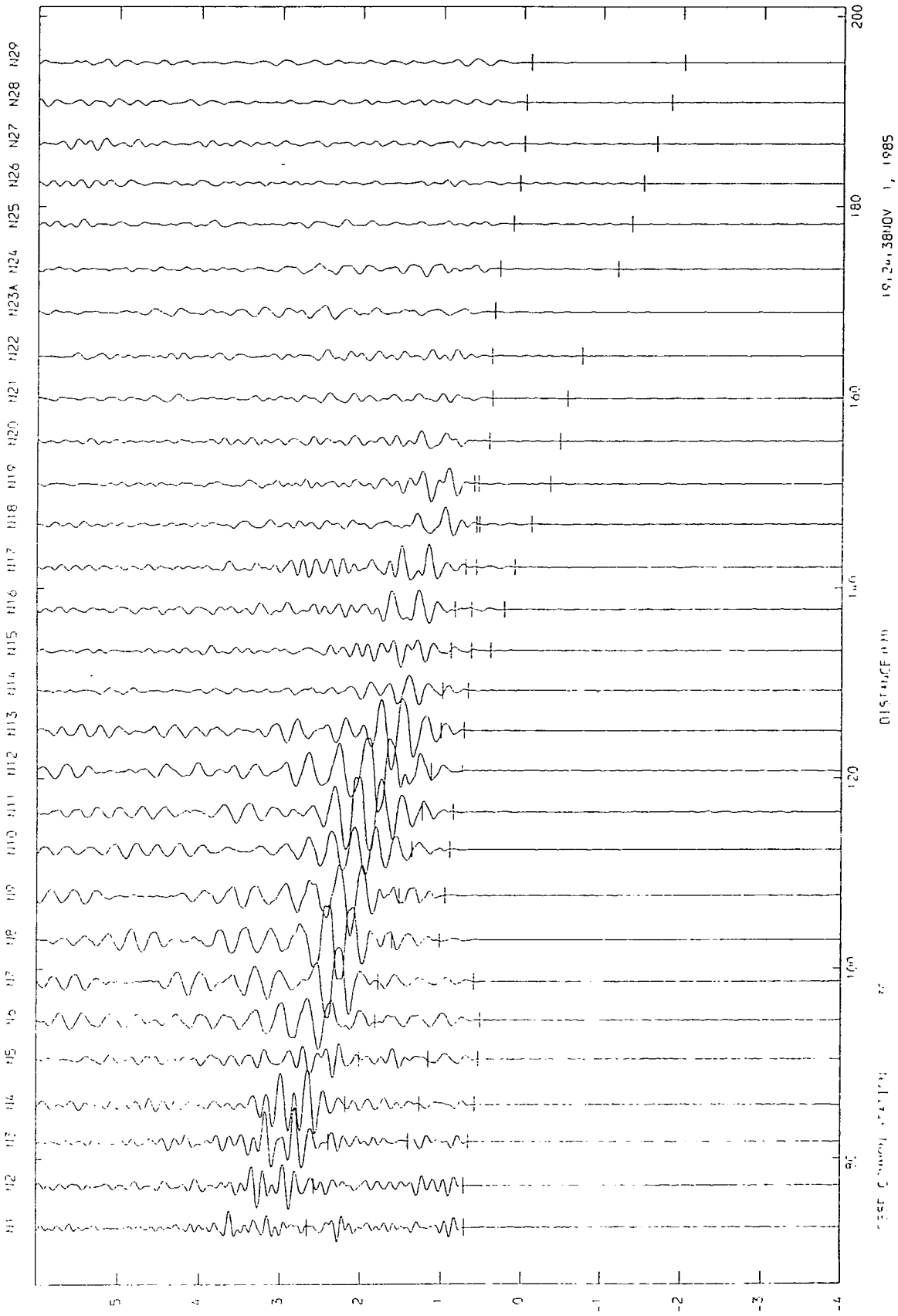
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0036858



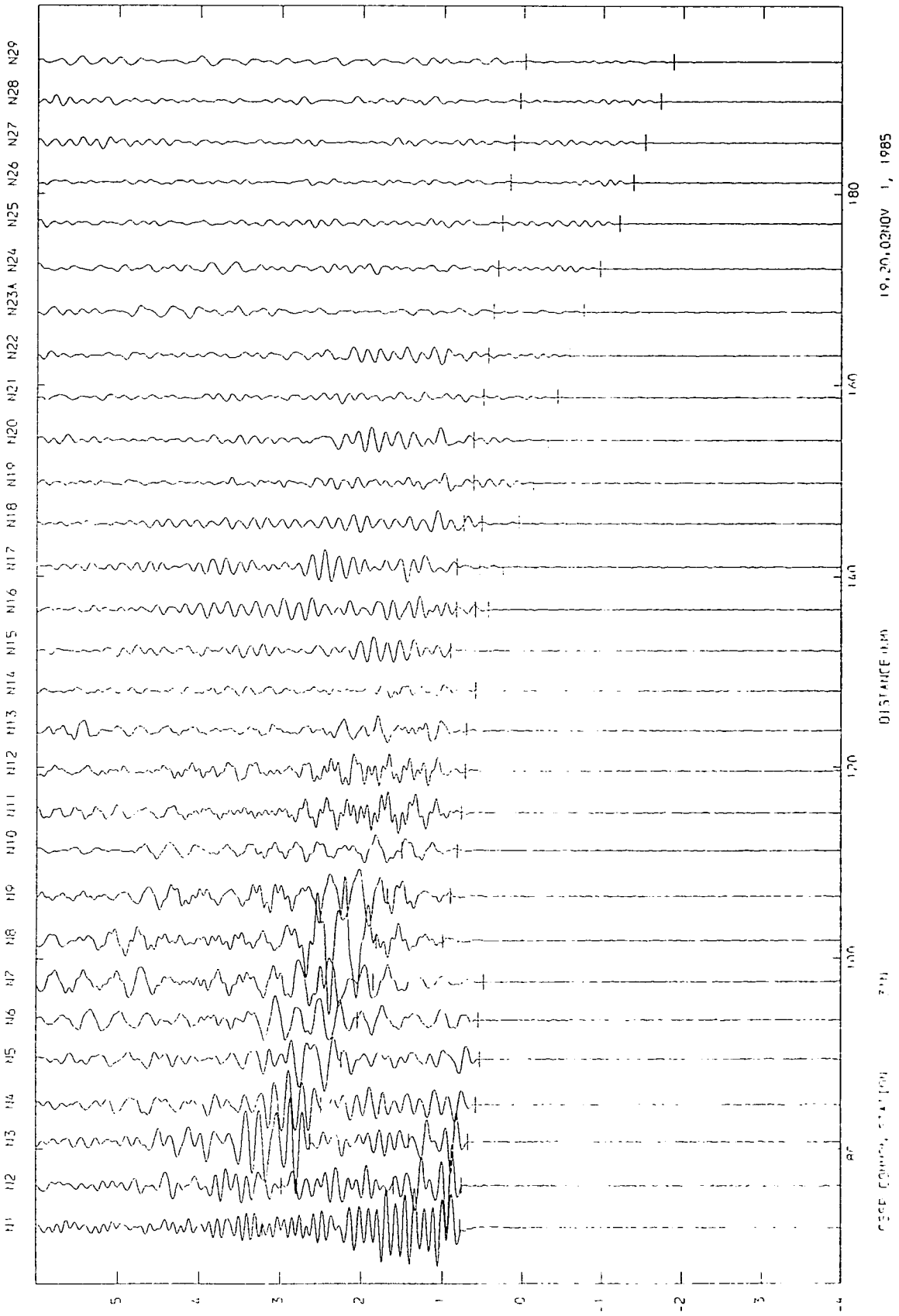
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0049207



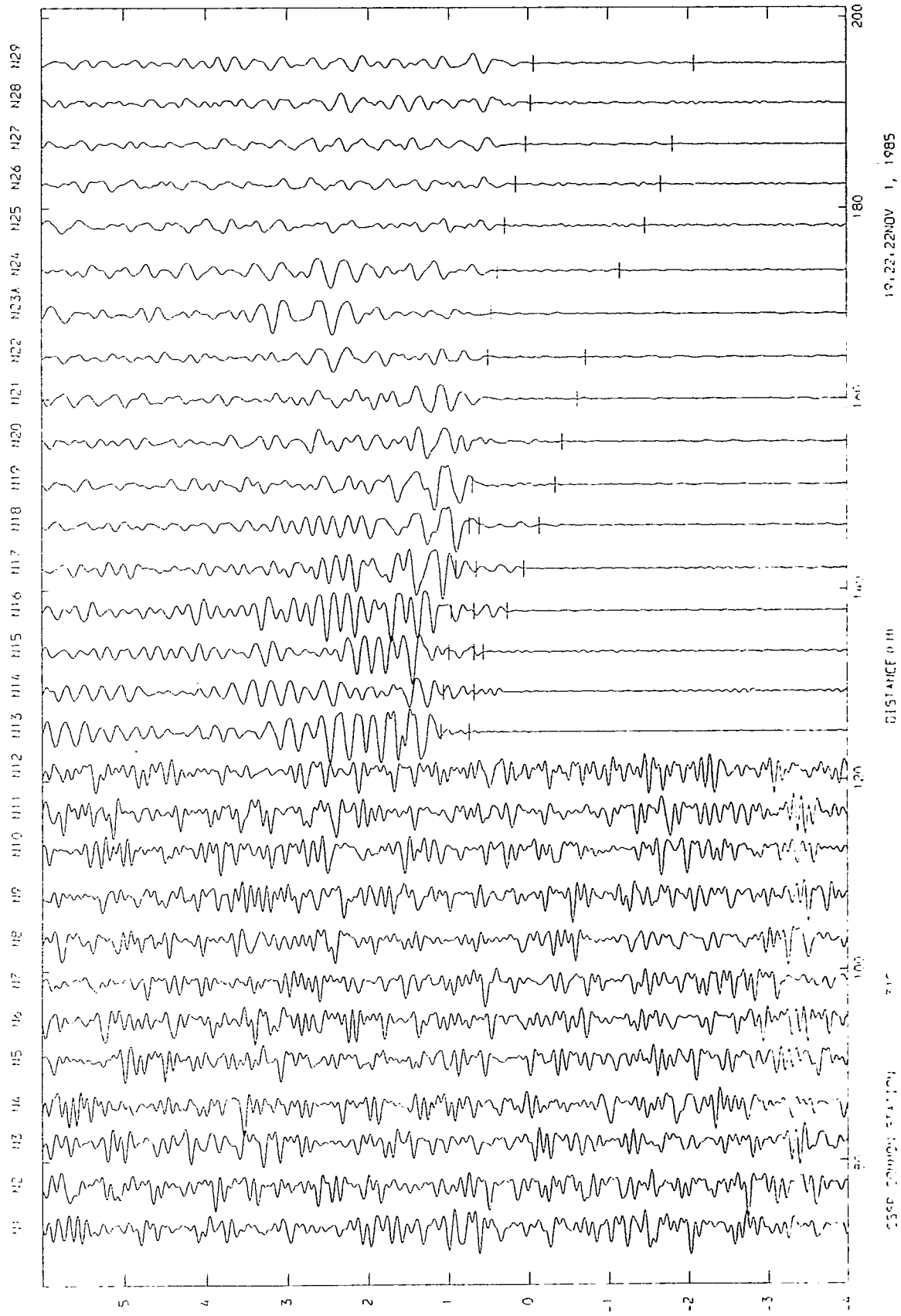
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0019584



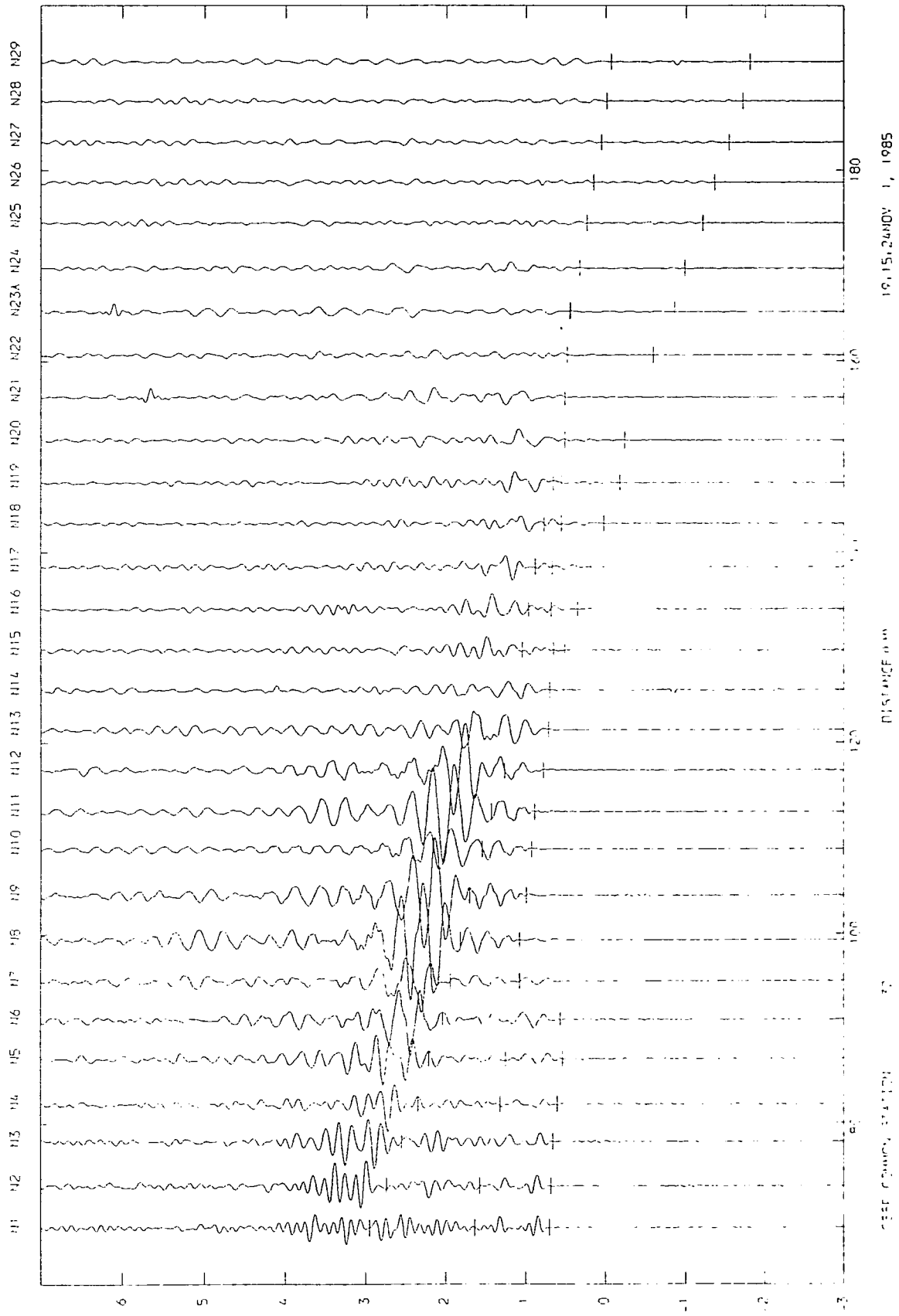
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0040565



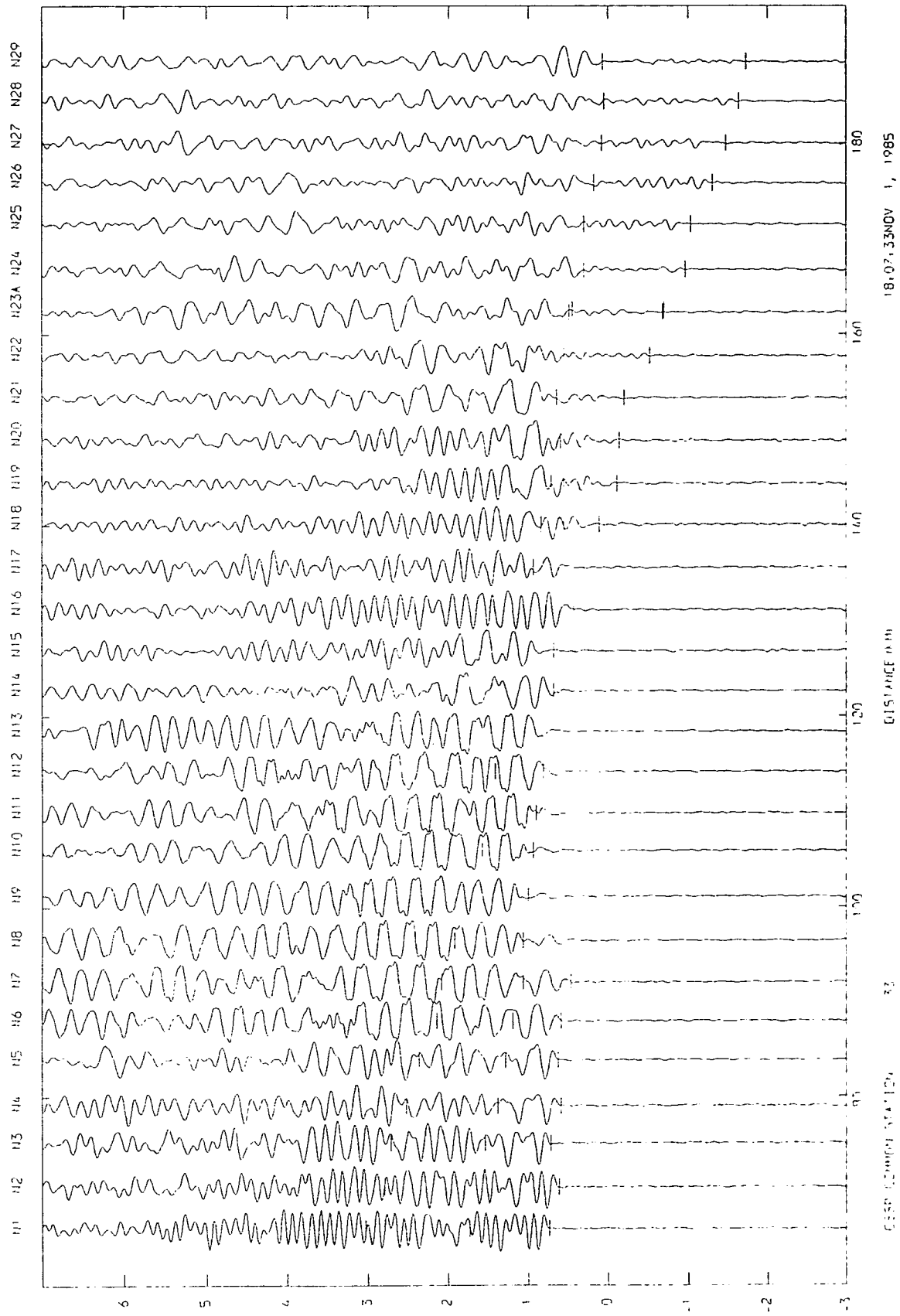
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0010941



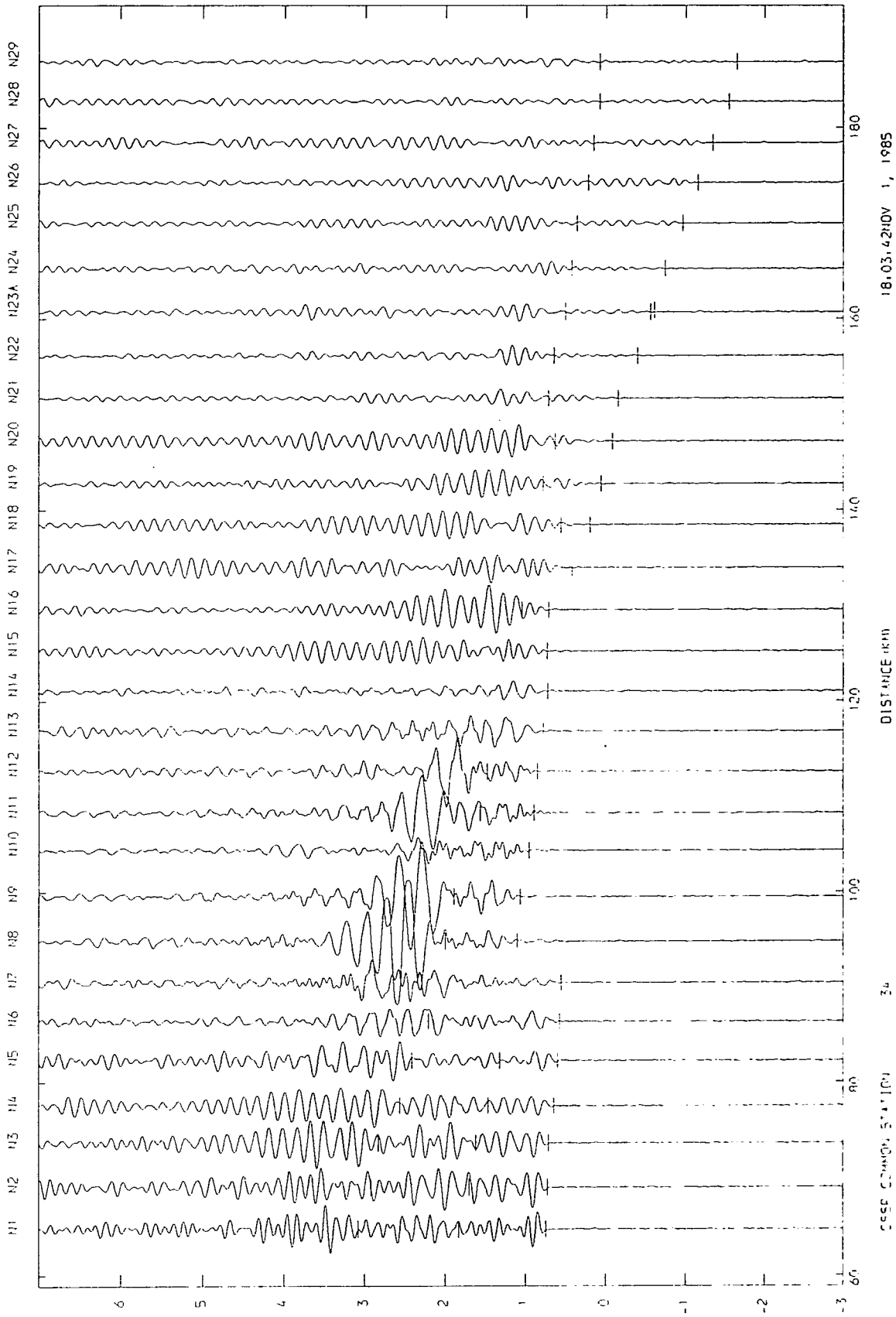
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0031364



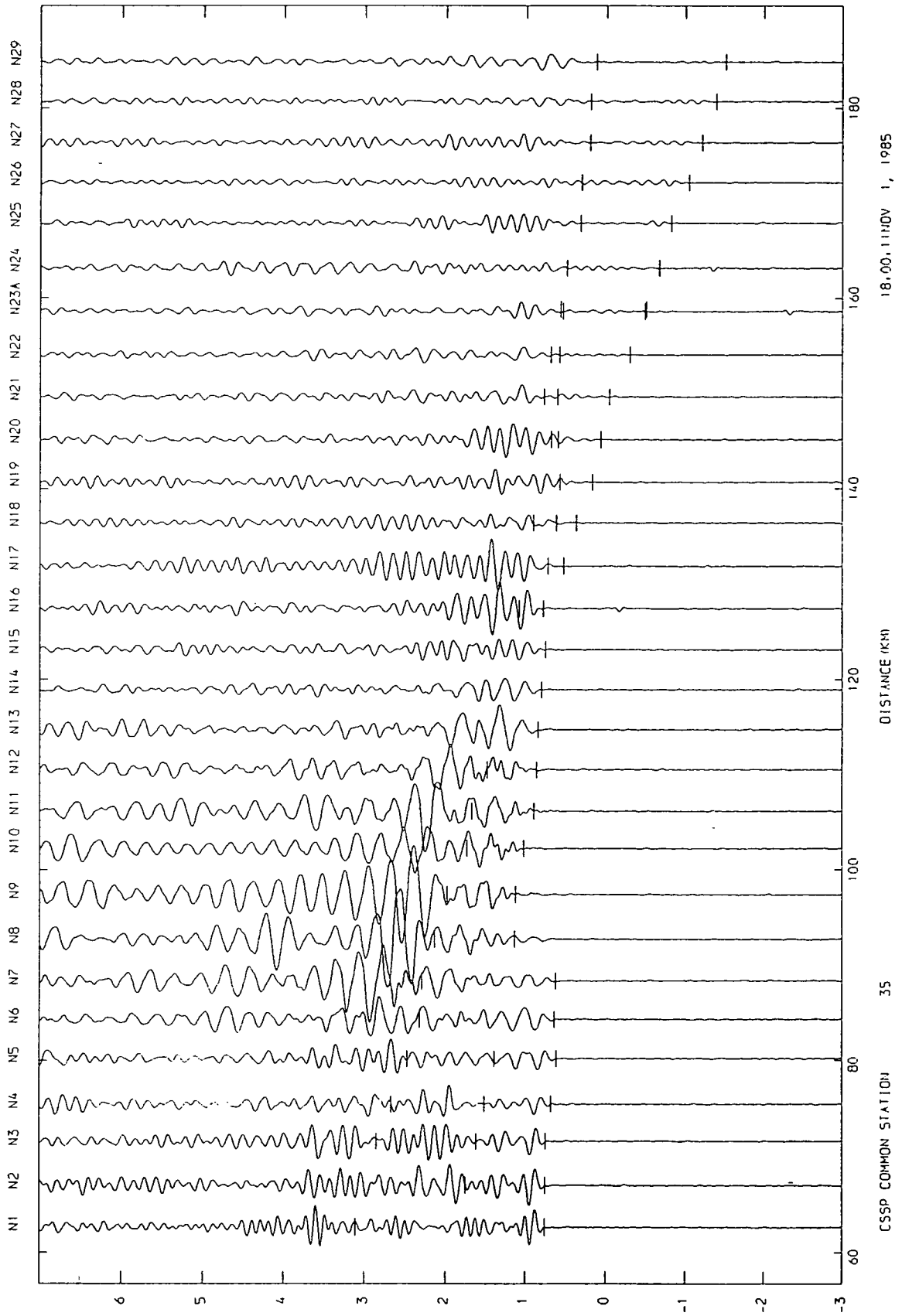
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0295985



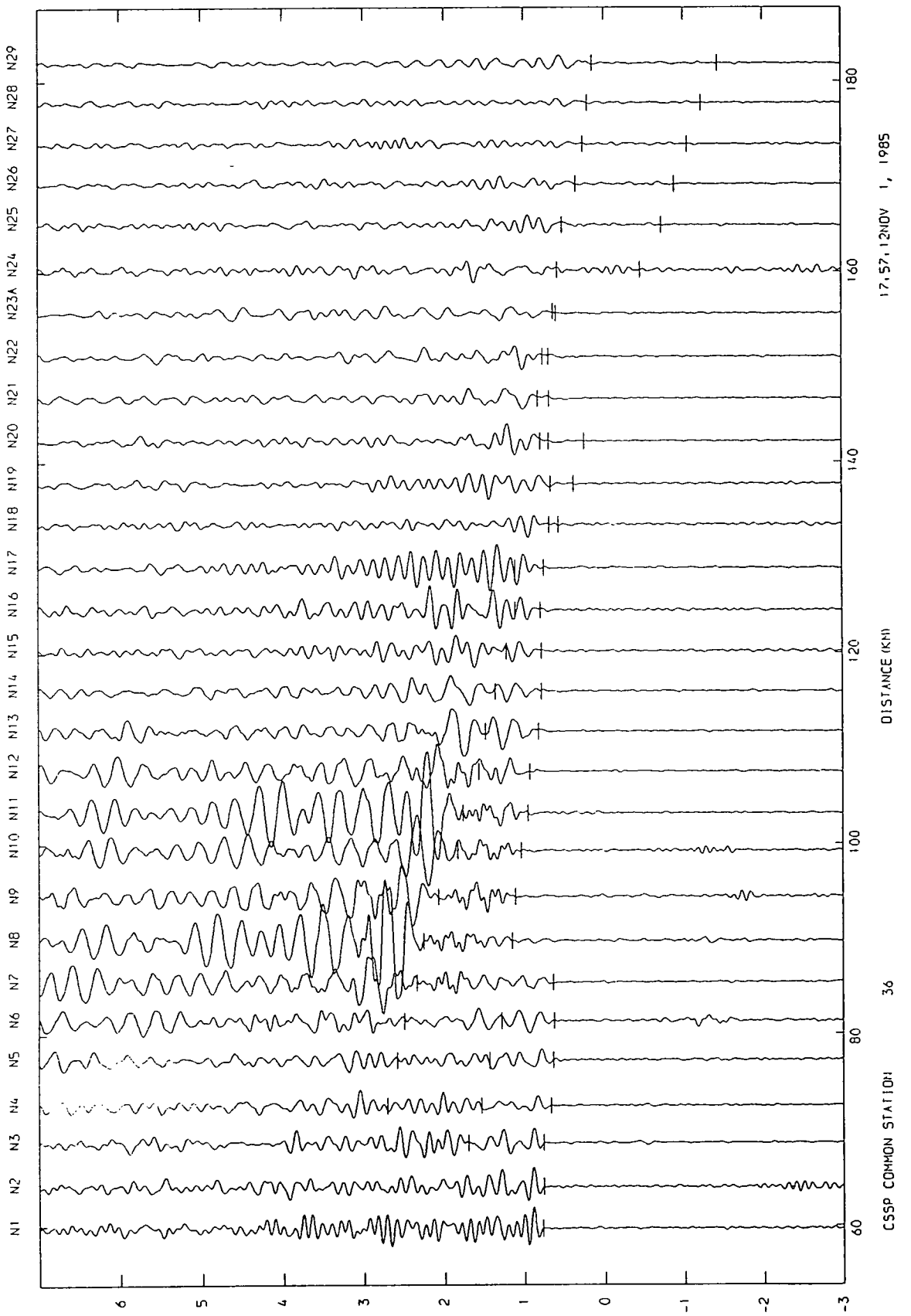
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0033536



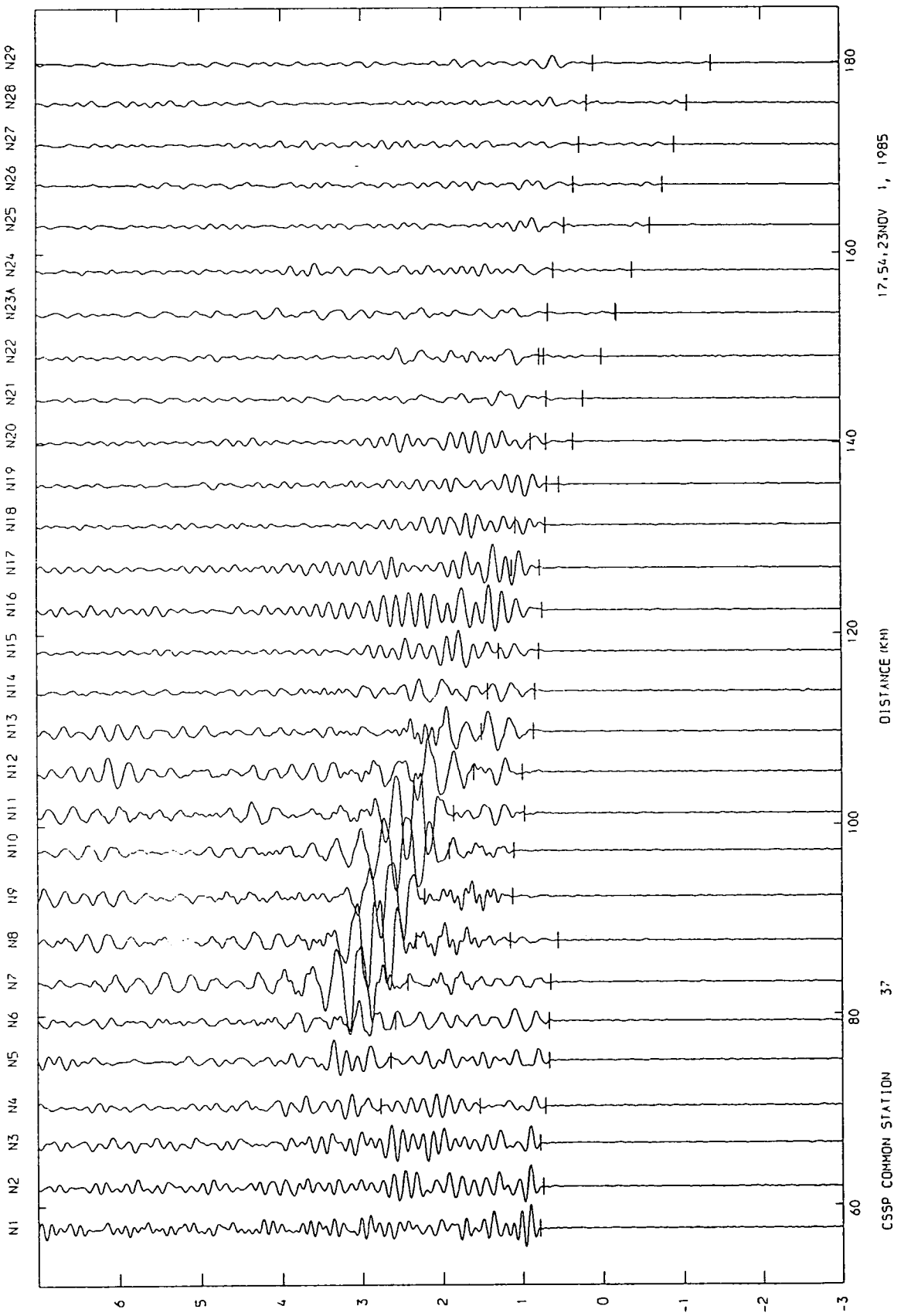
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0032396



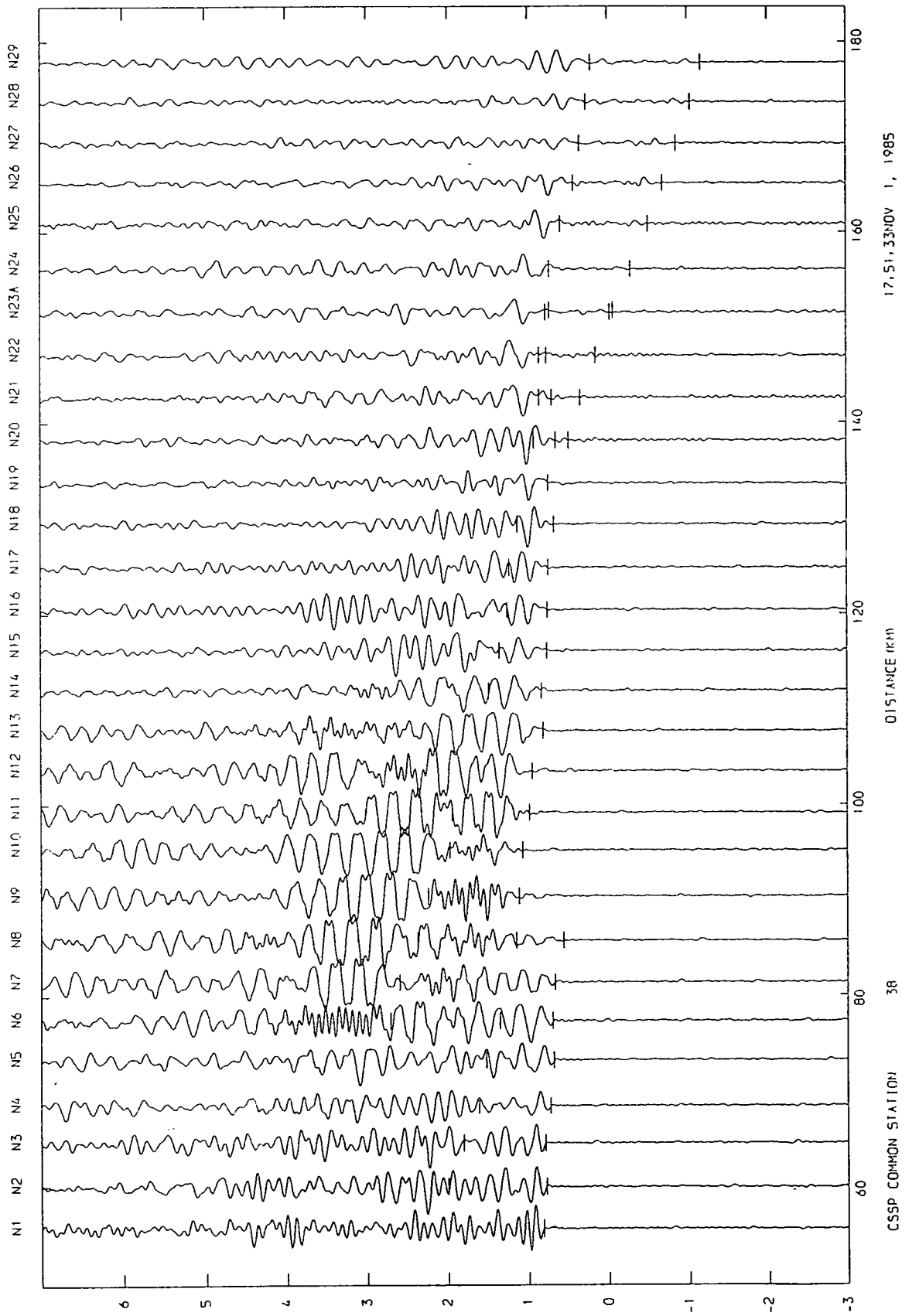
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0046667



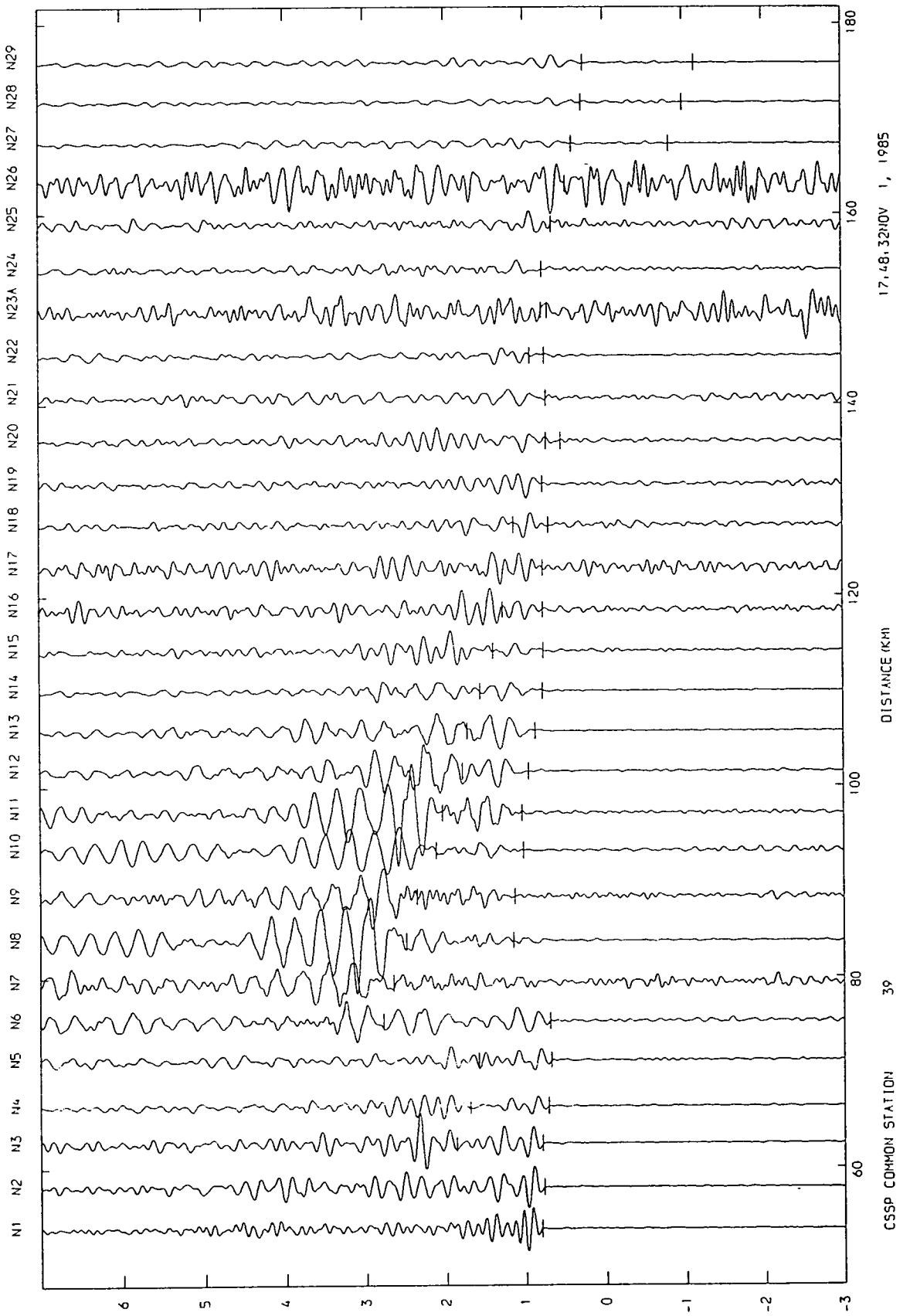
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0024037



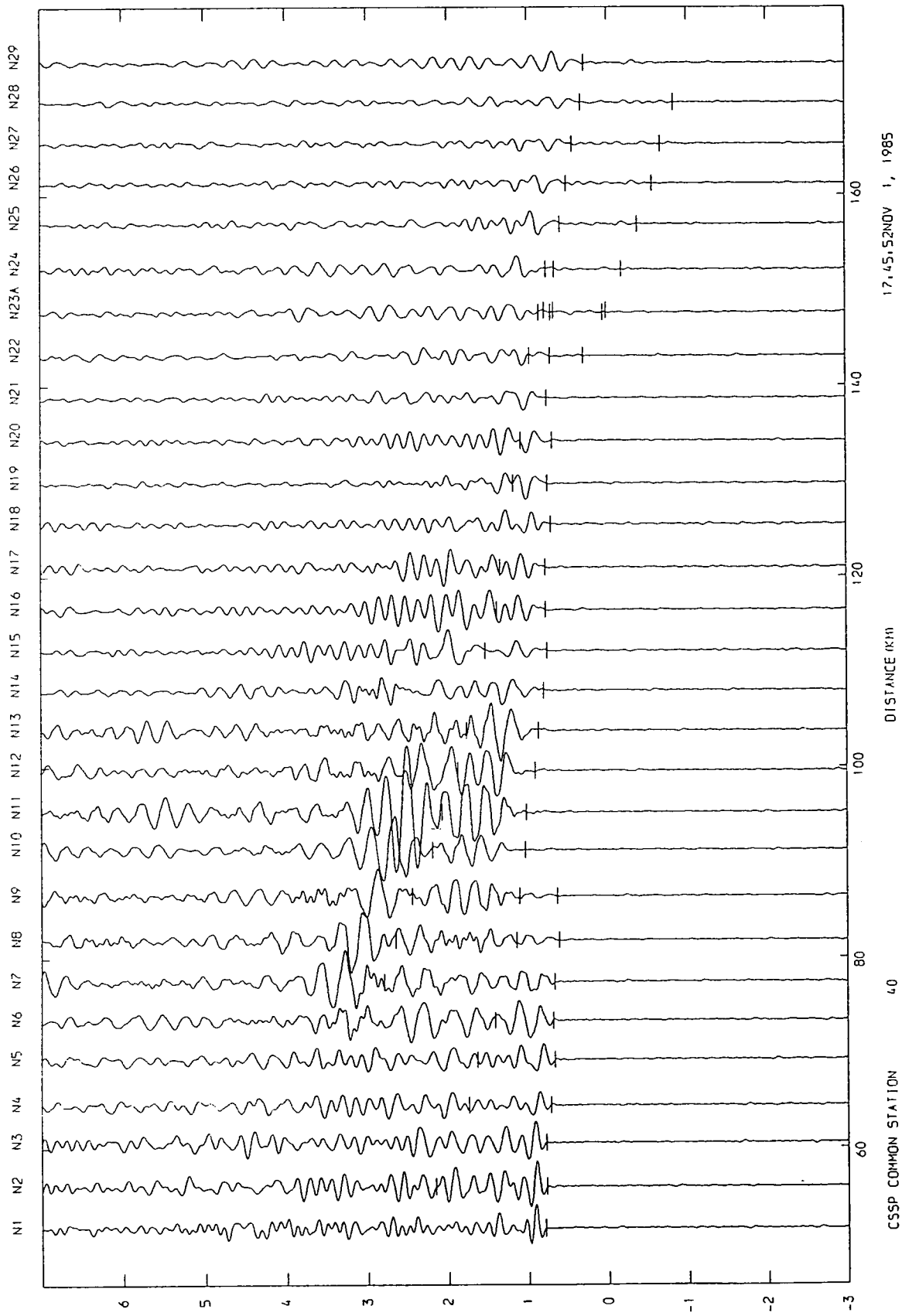
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0010358



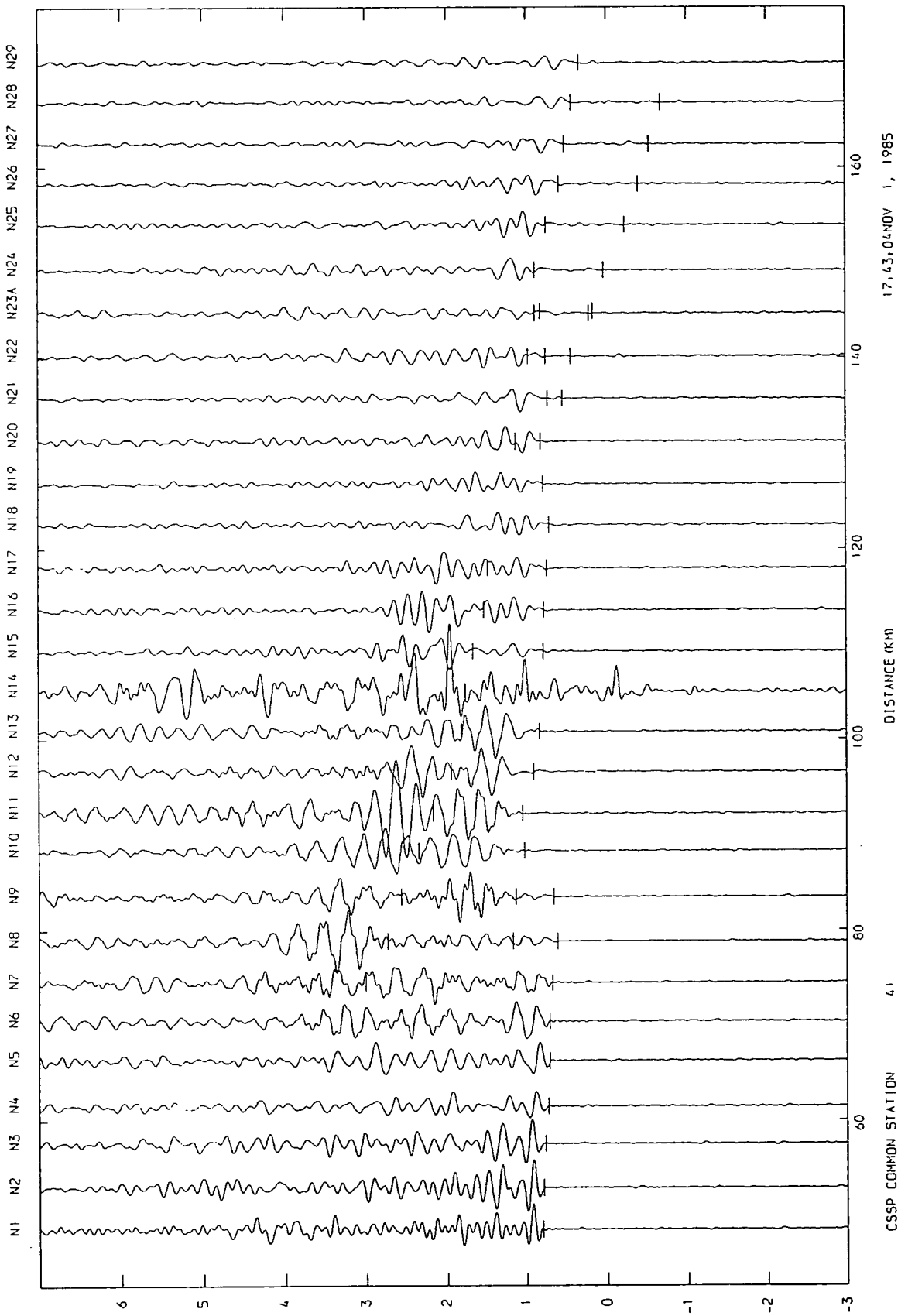
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0026730



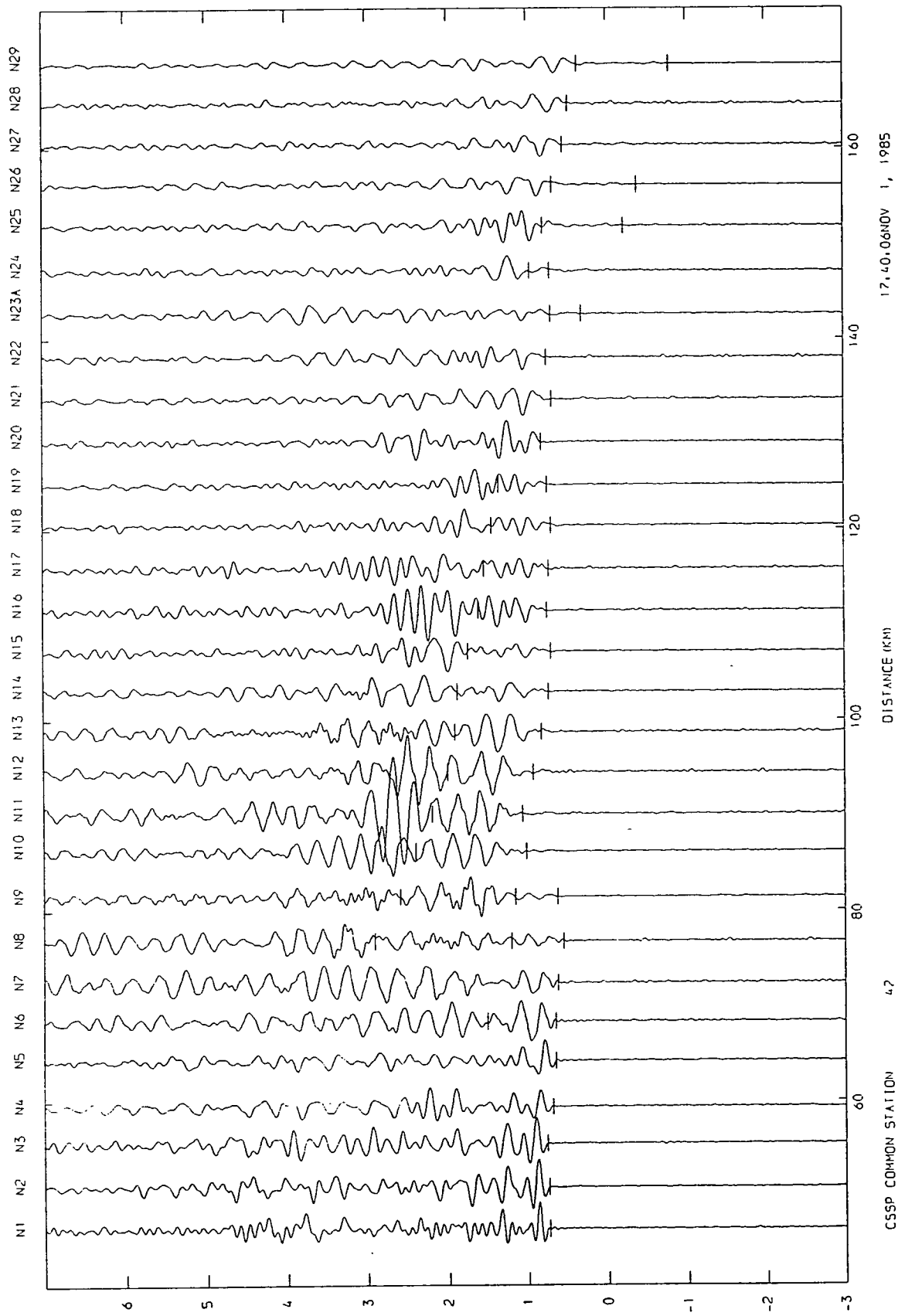
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0043230



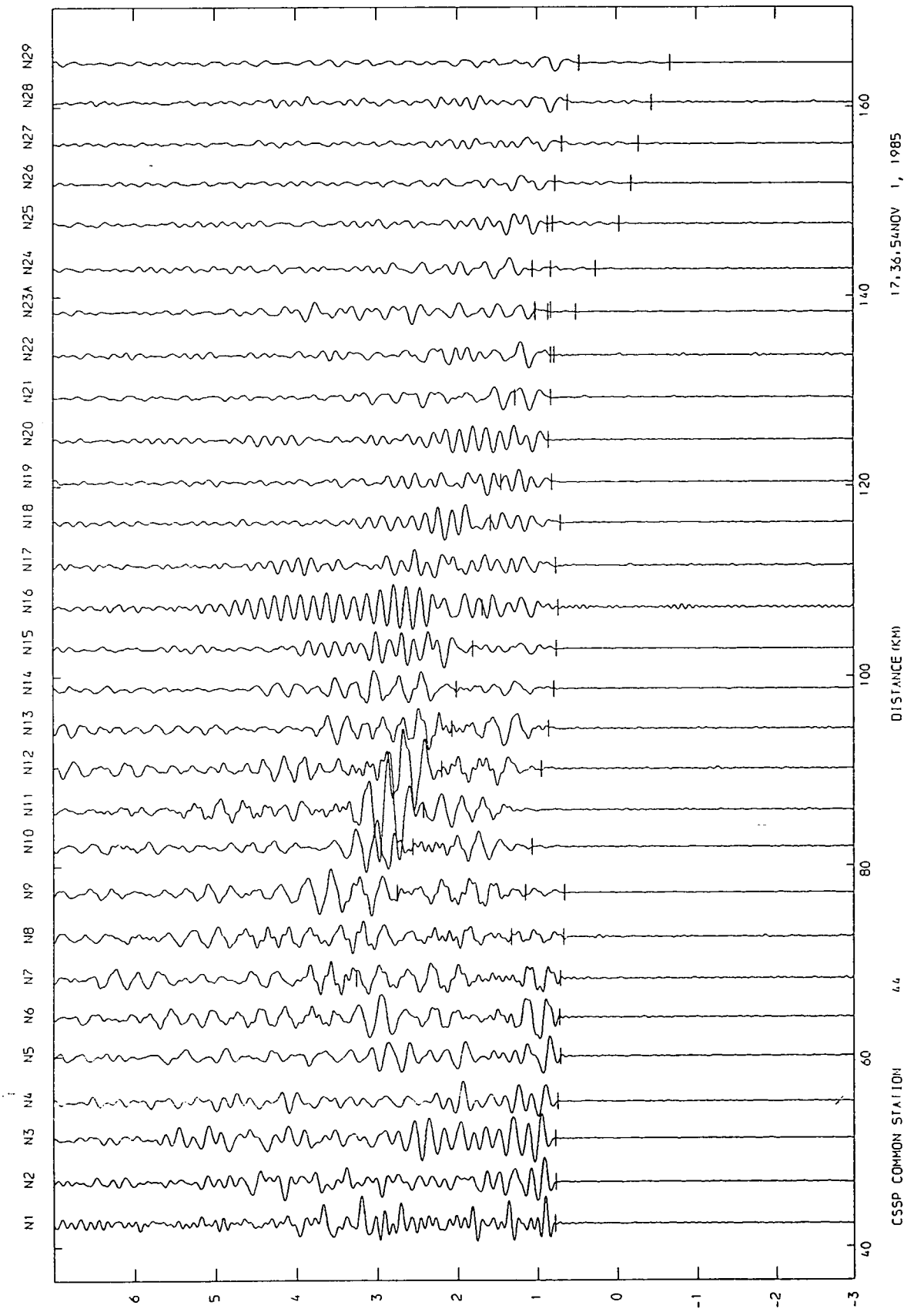
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0040997



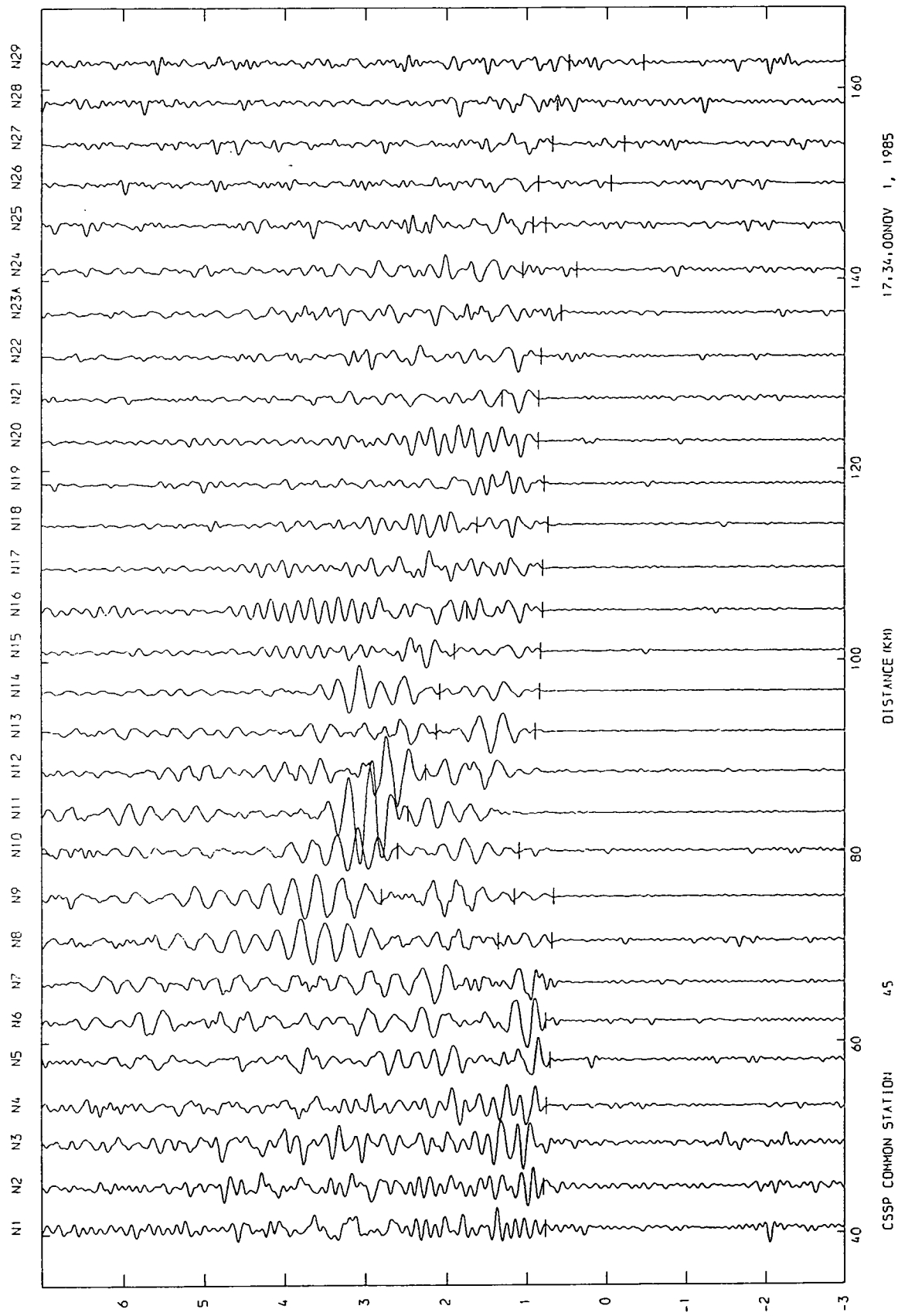
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.006066B



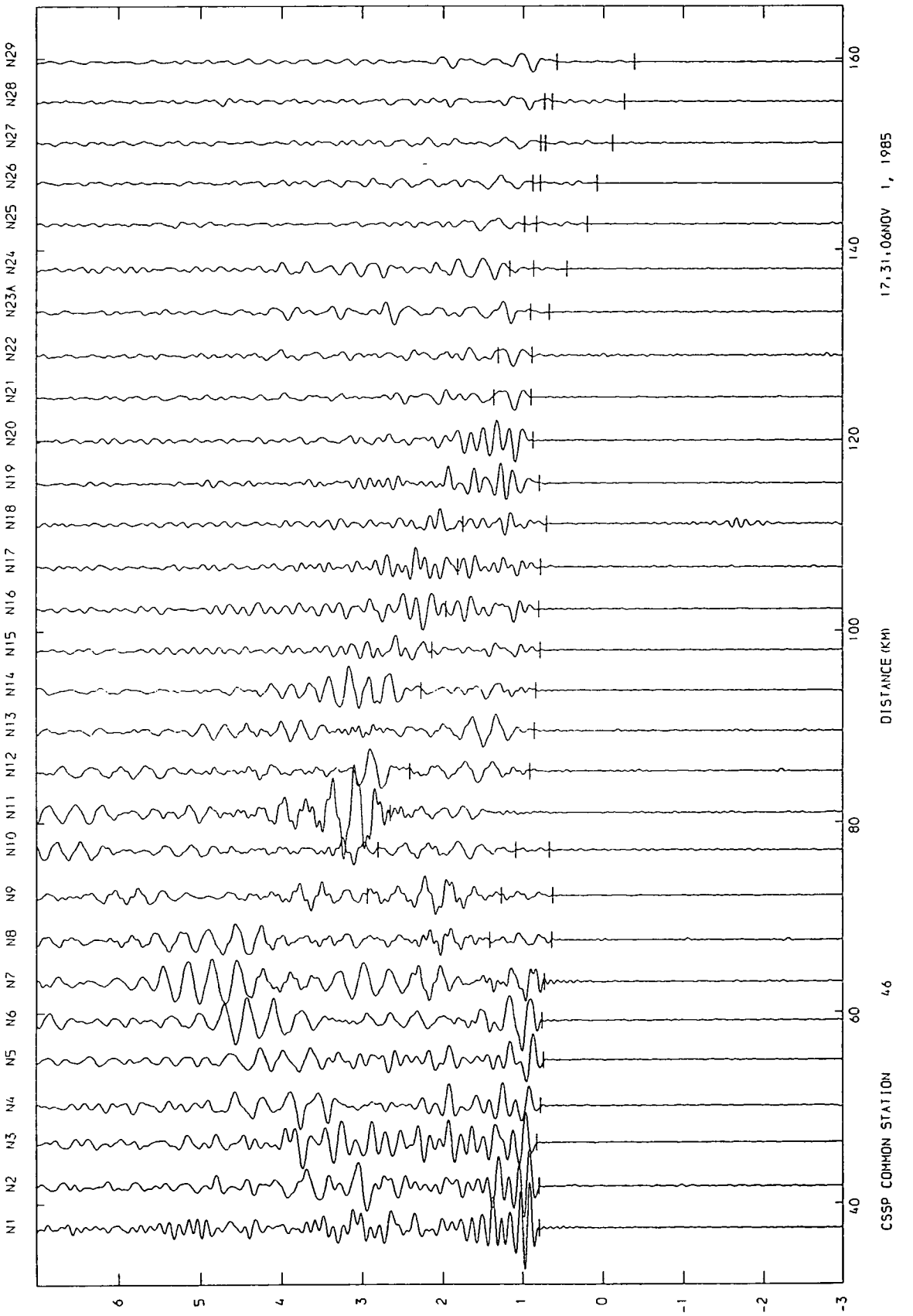
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0040656



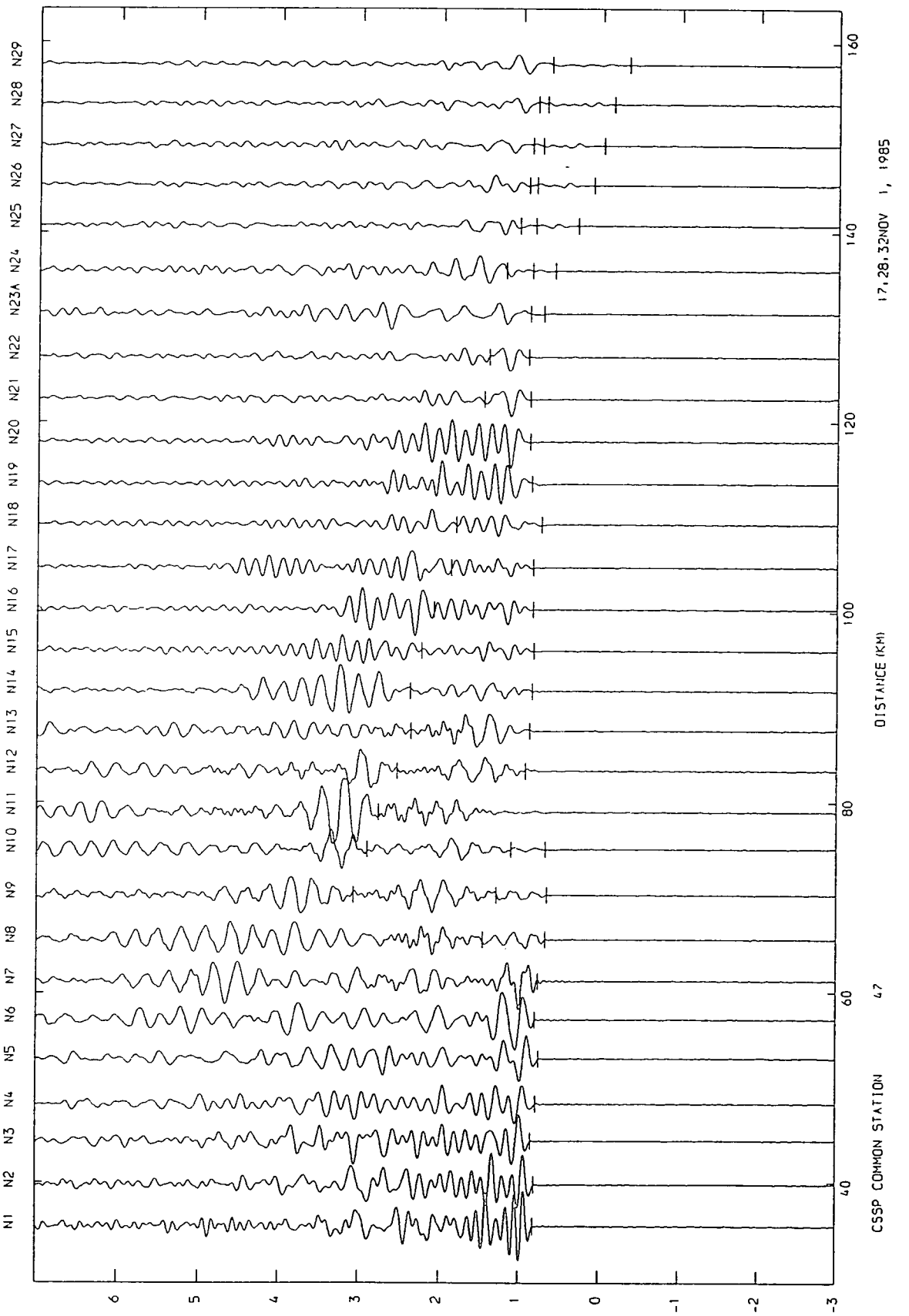
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0044908



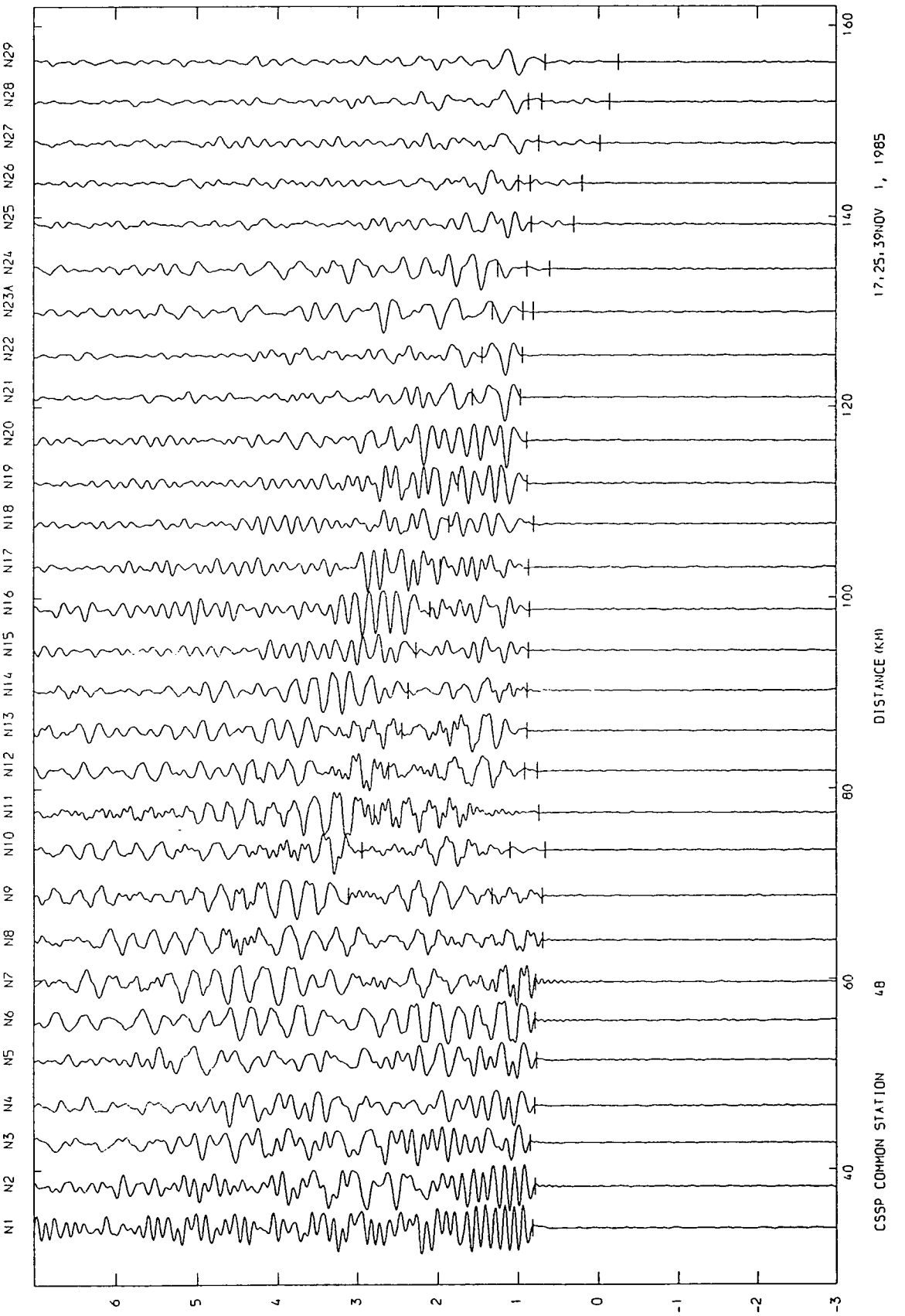
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0032442



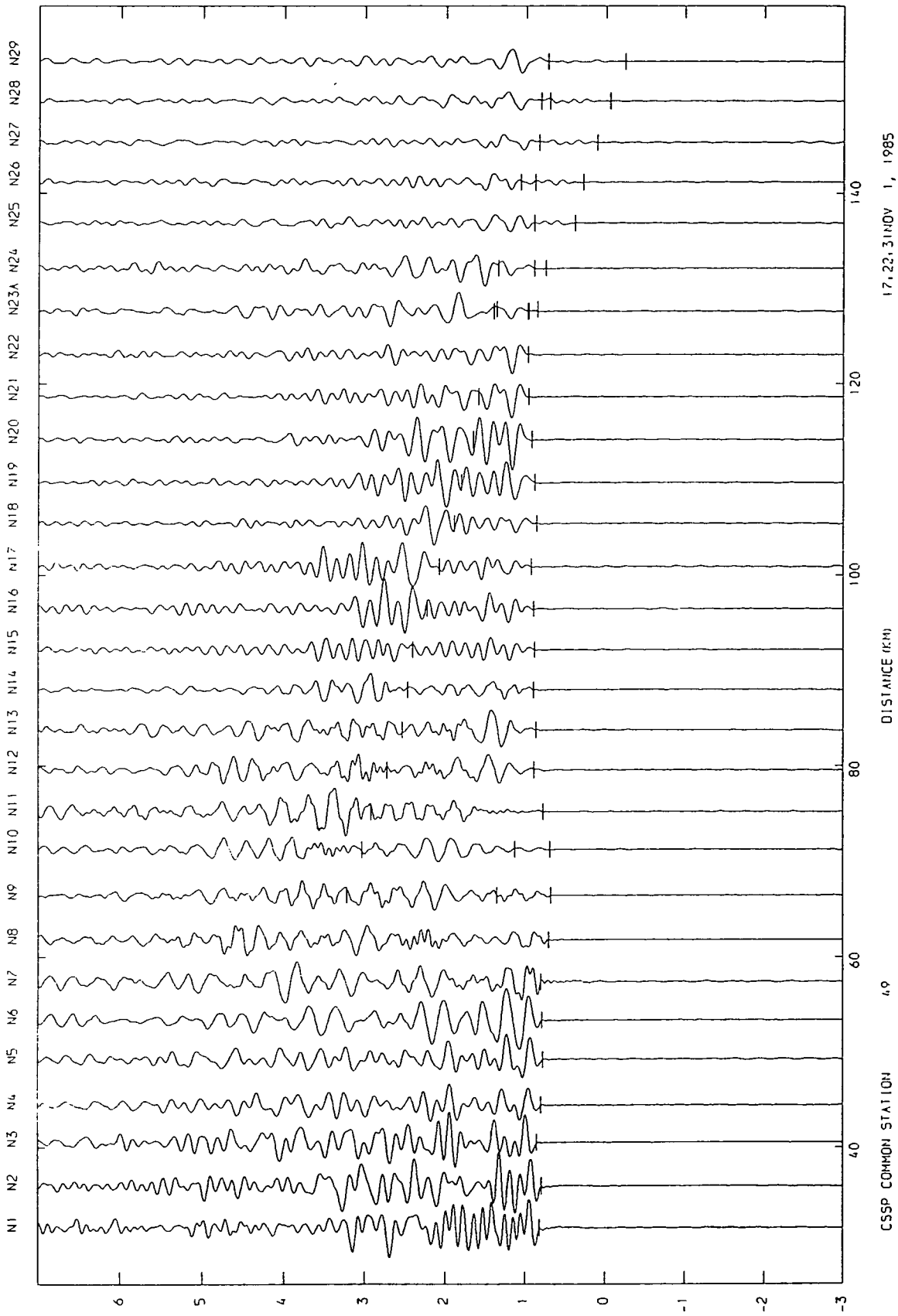
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0025013



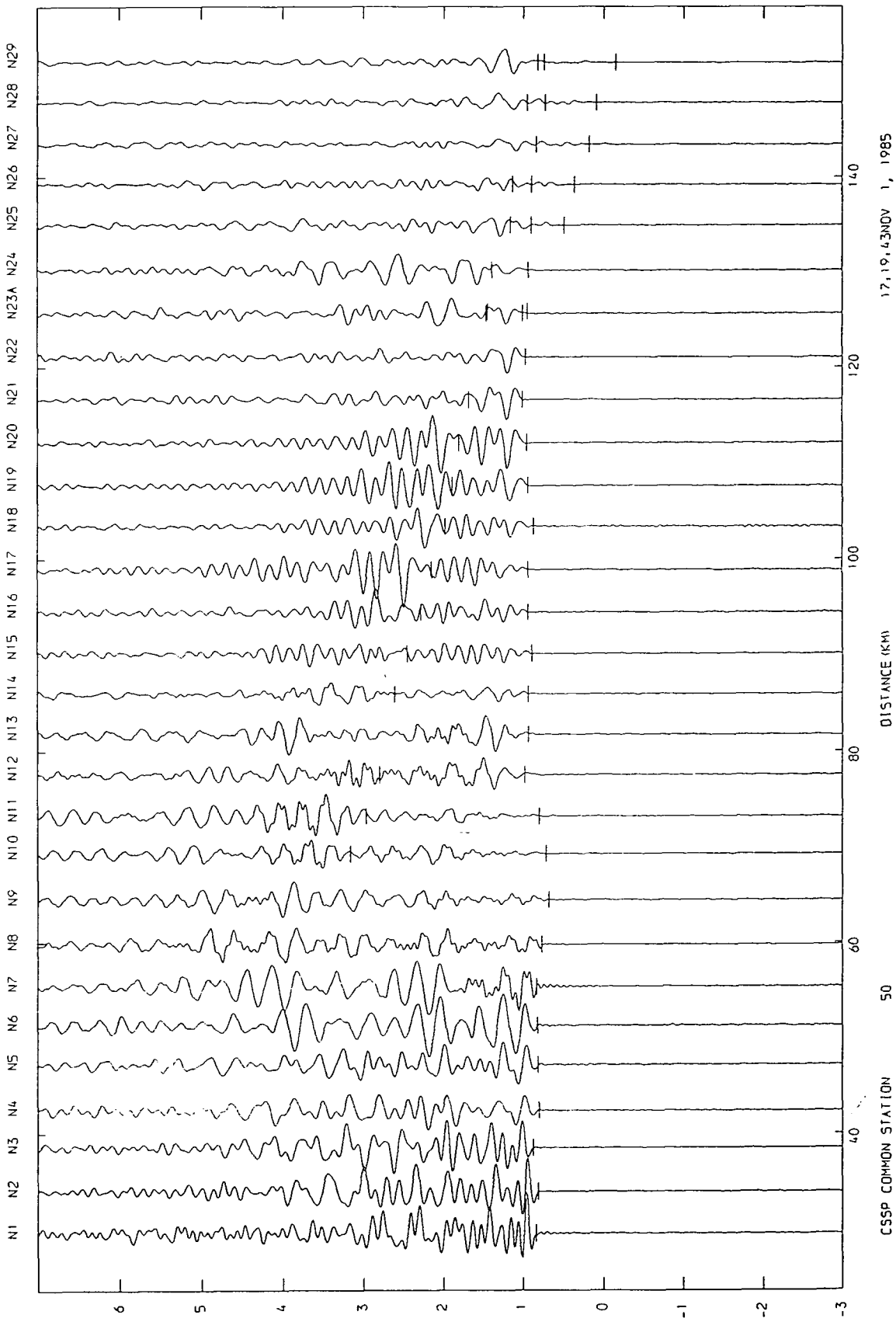
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0009417



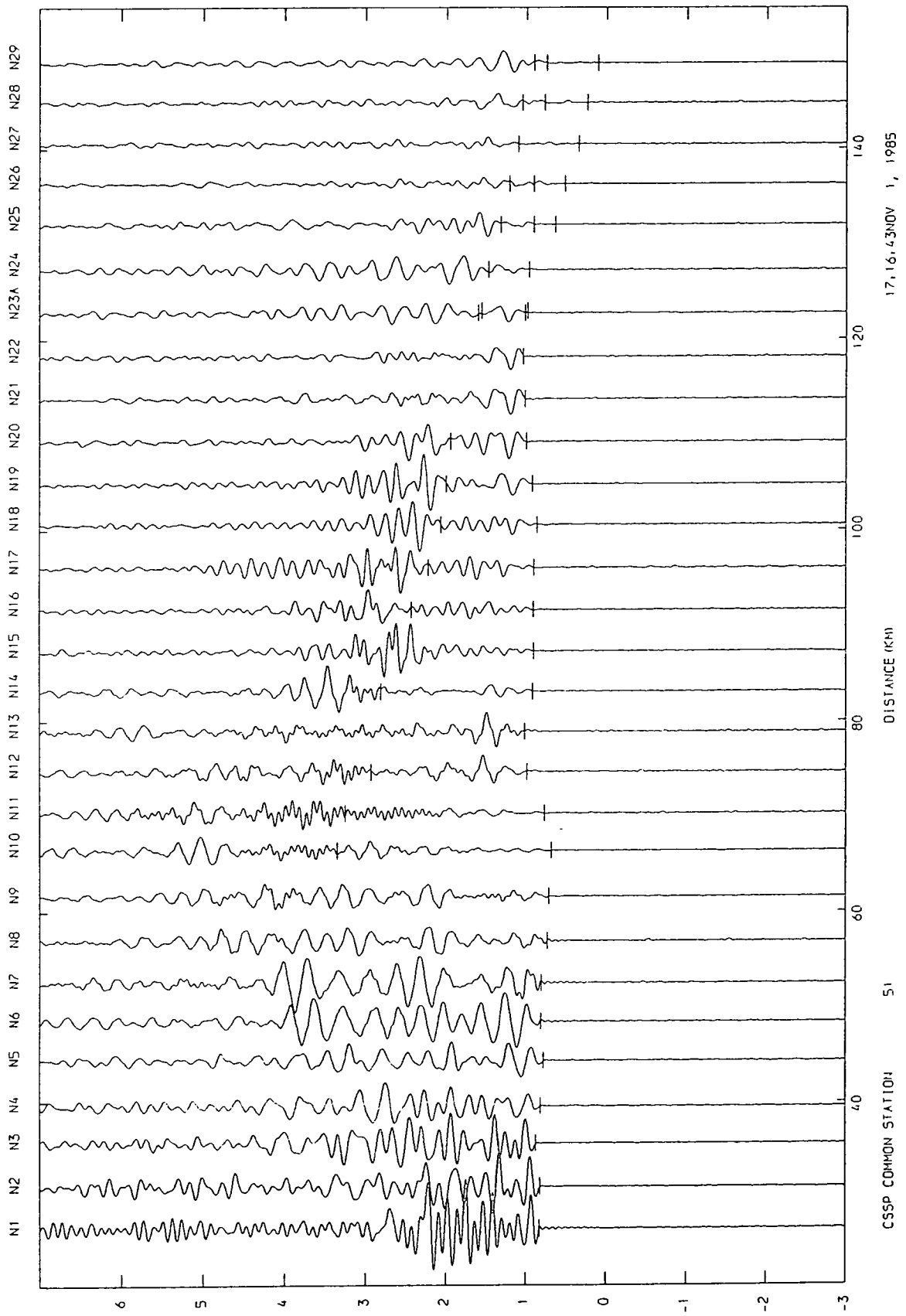
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0046085



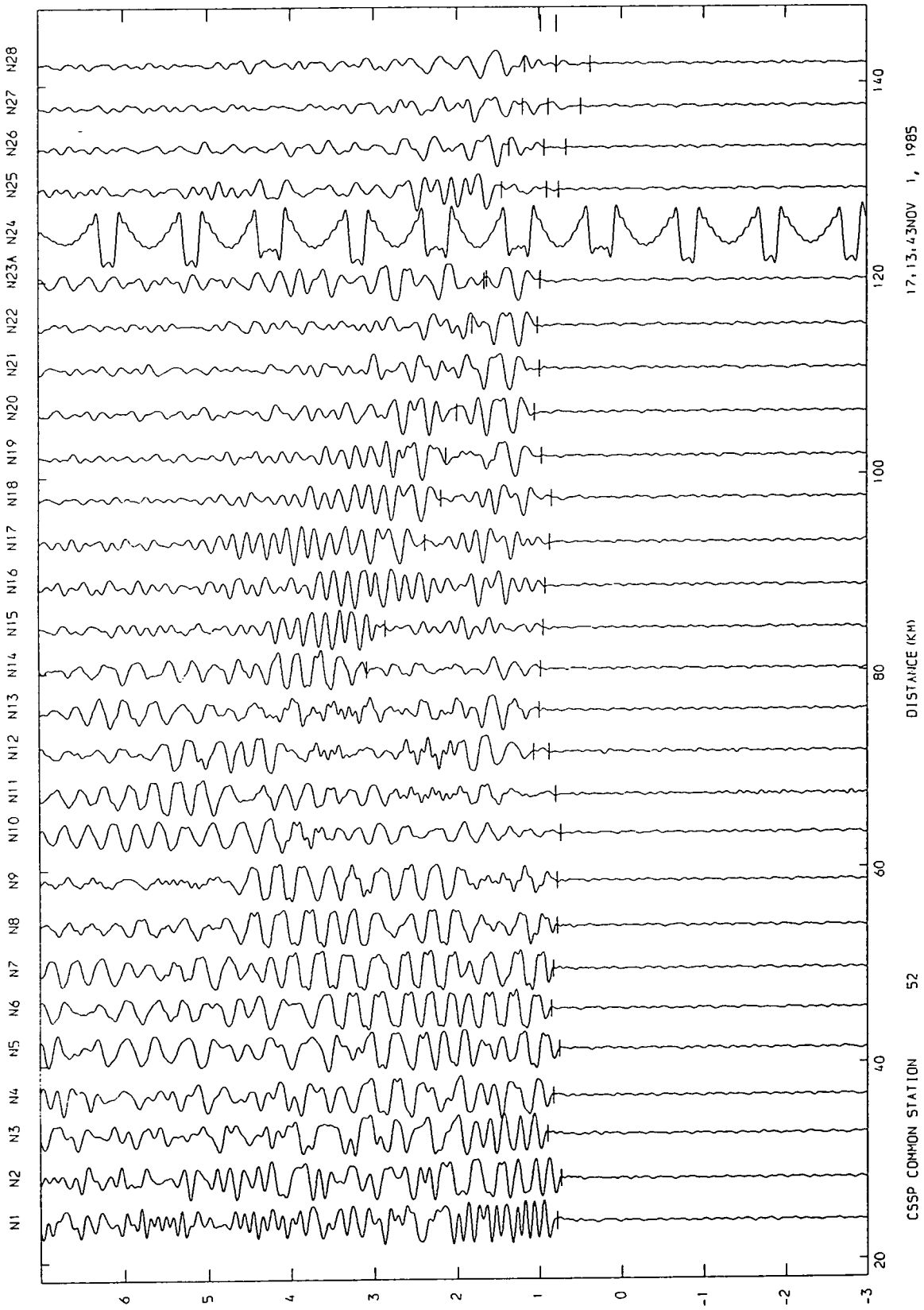
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0038645



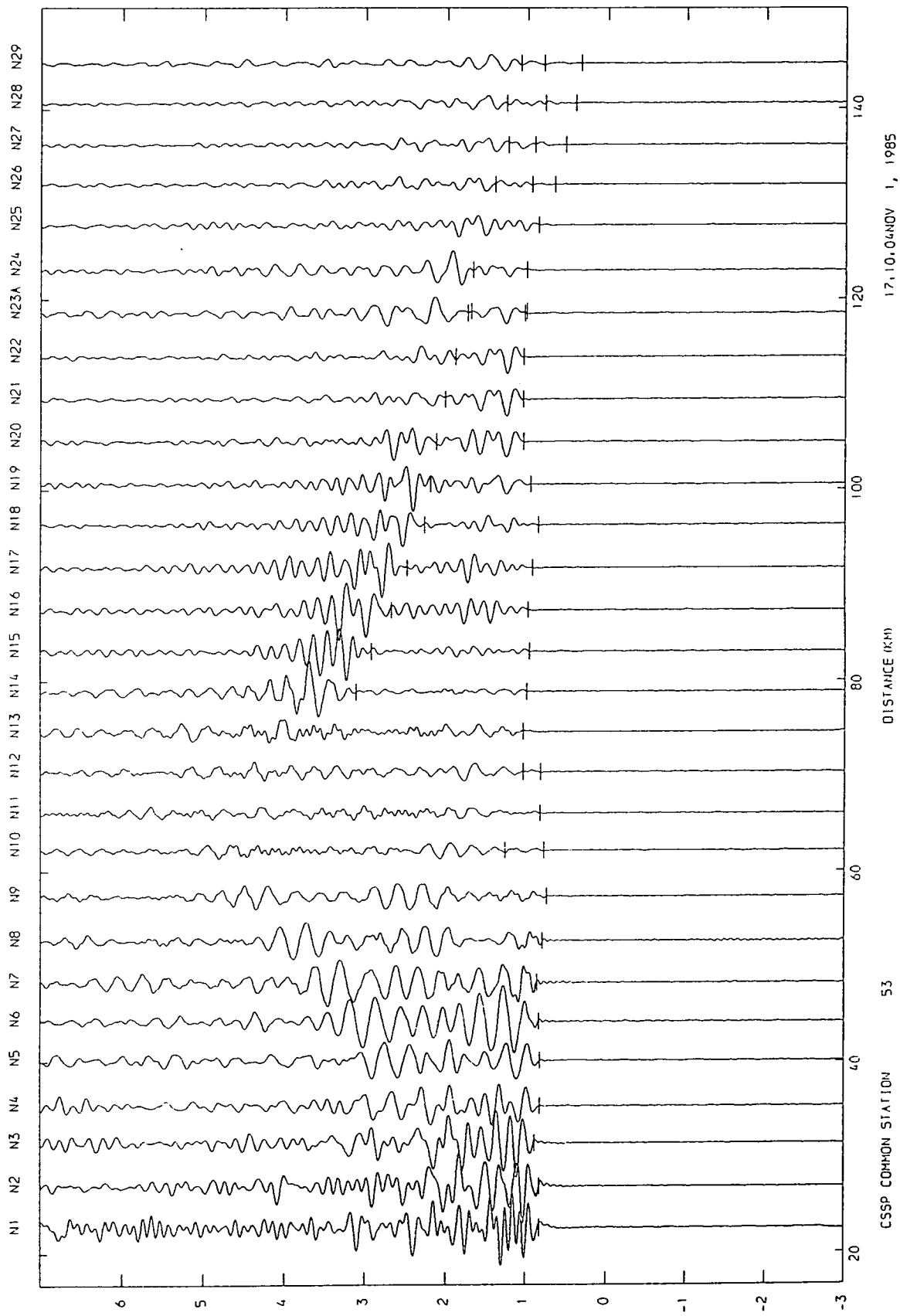
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0024572



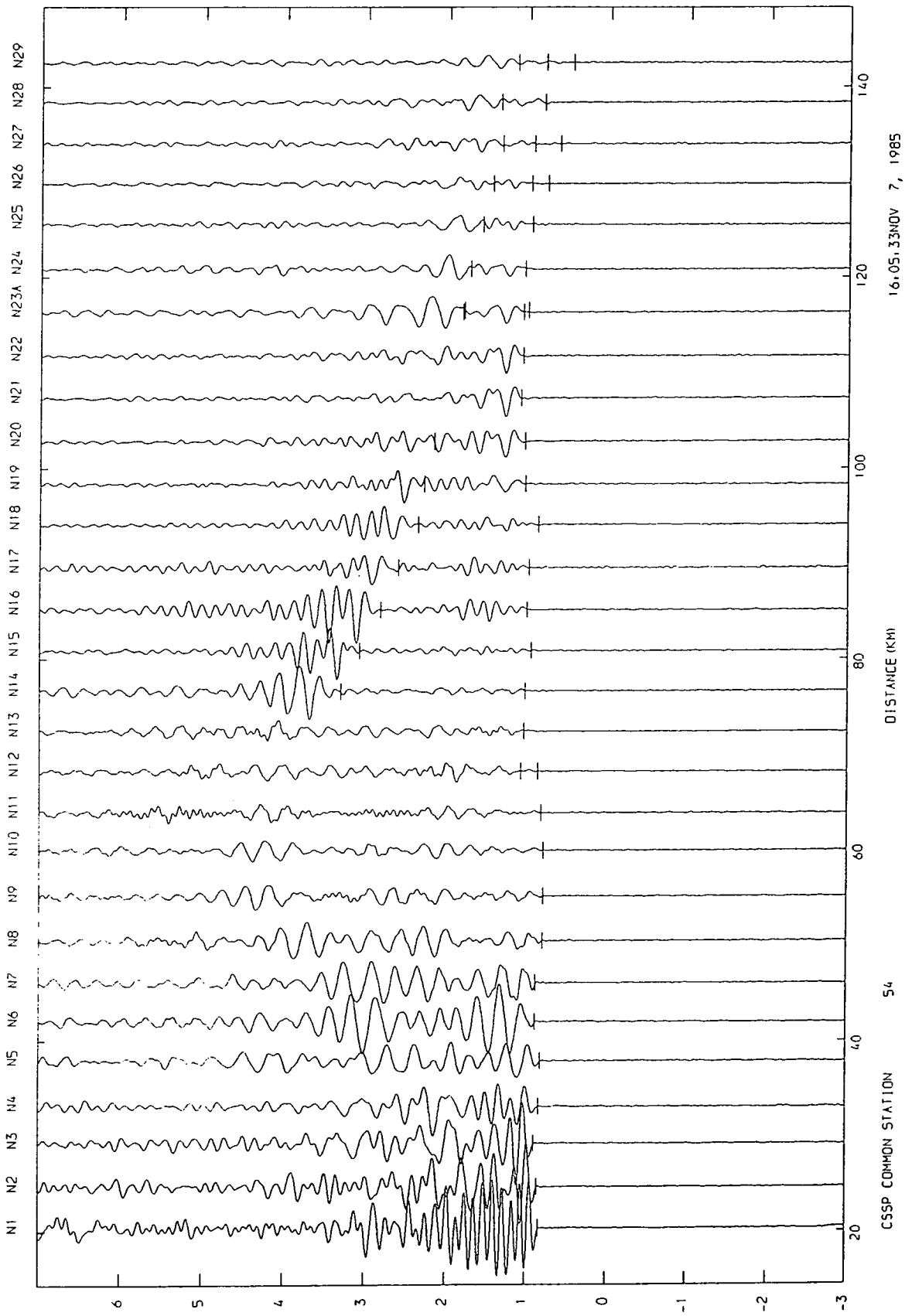
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0219439



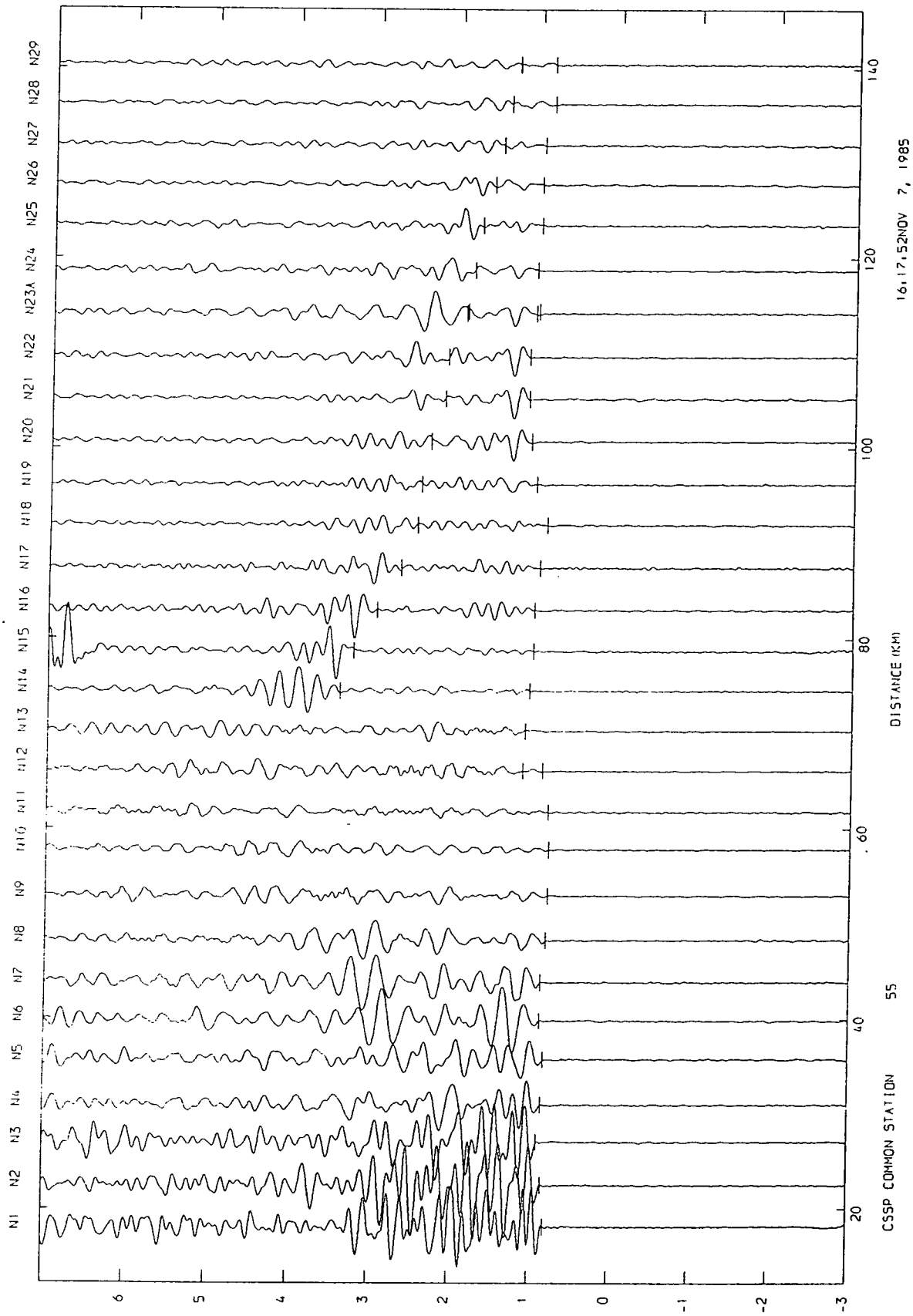
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0025344



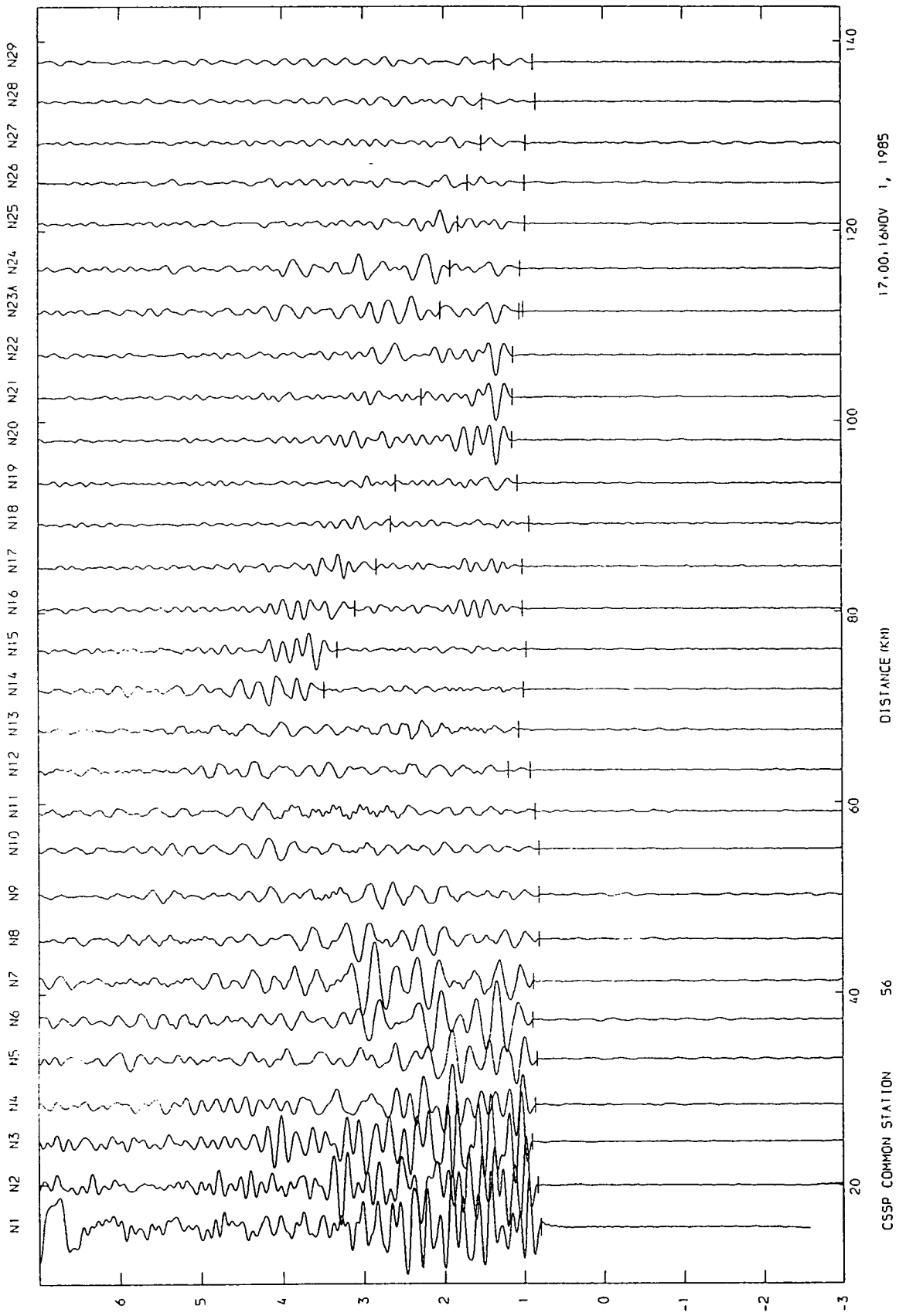
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0026785



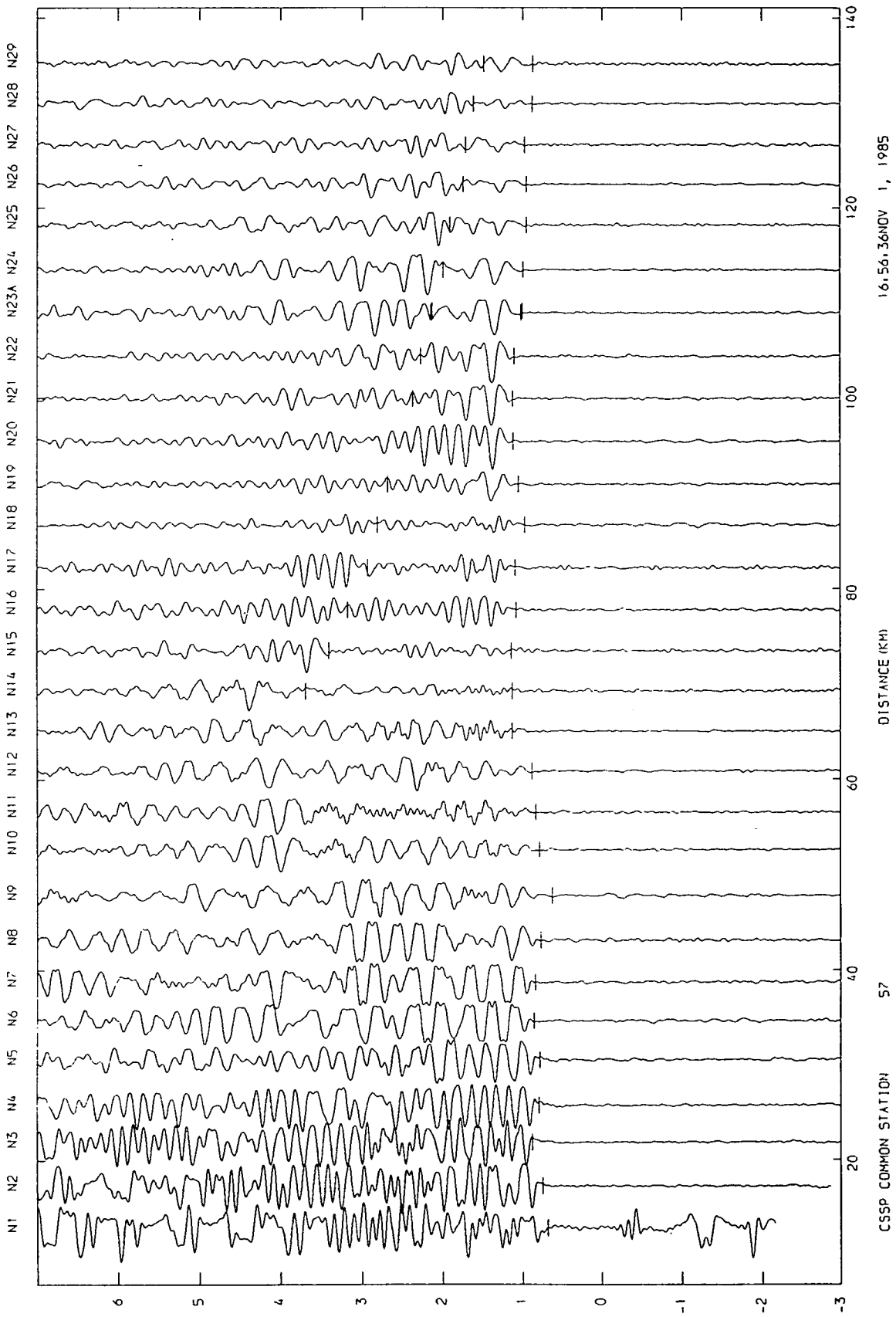
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0026915



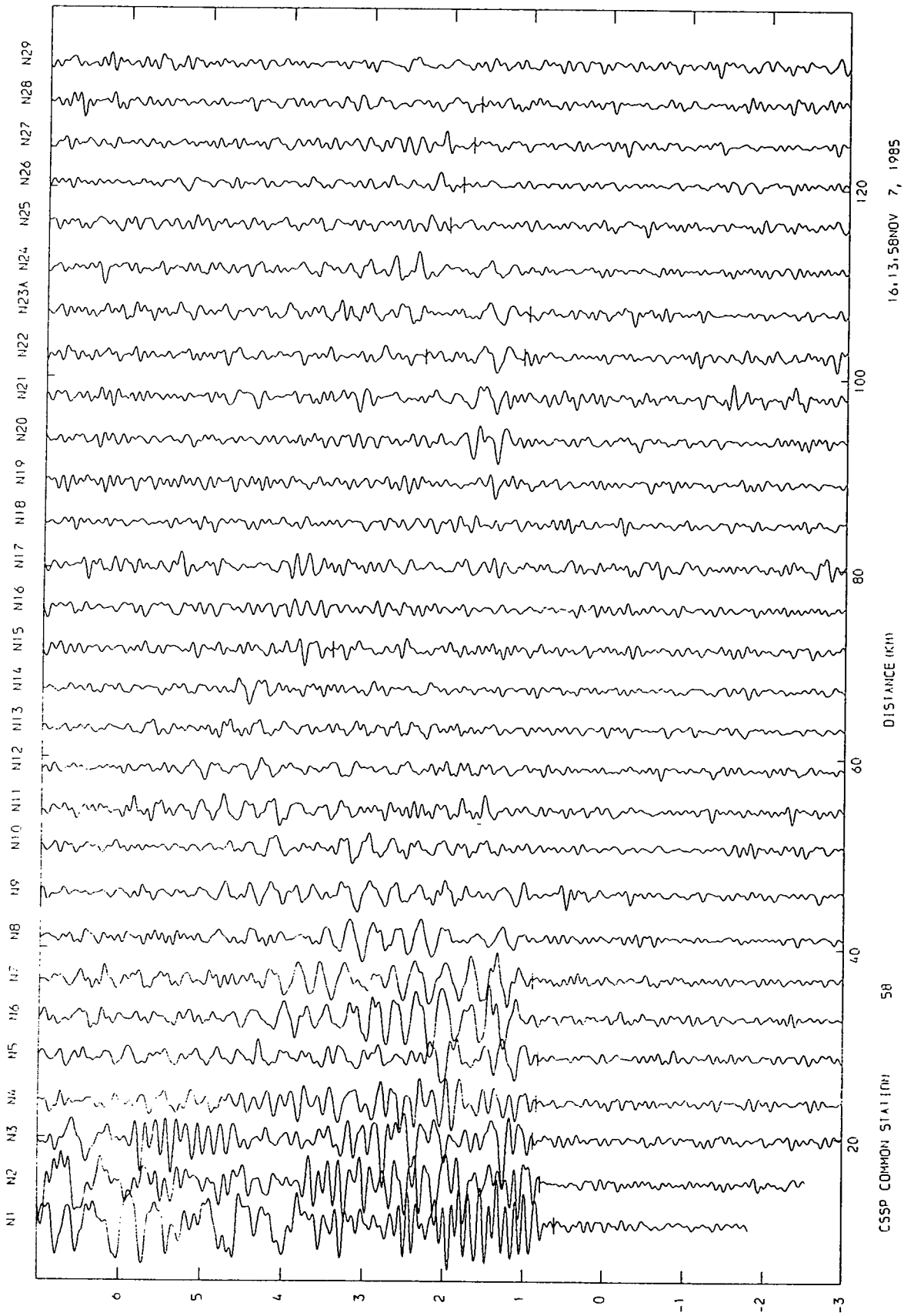
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0023138



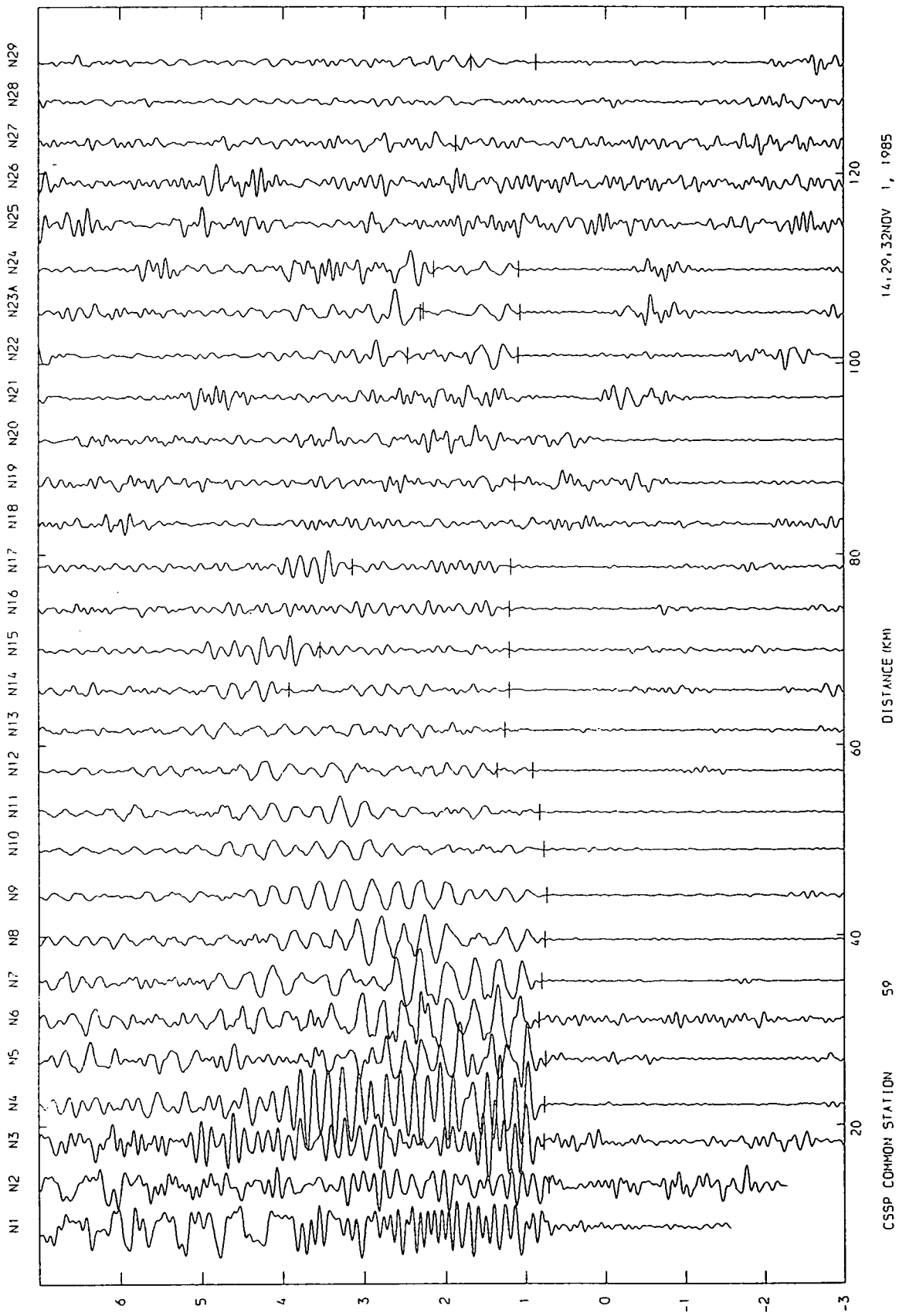
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0009086



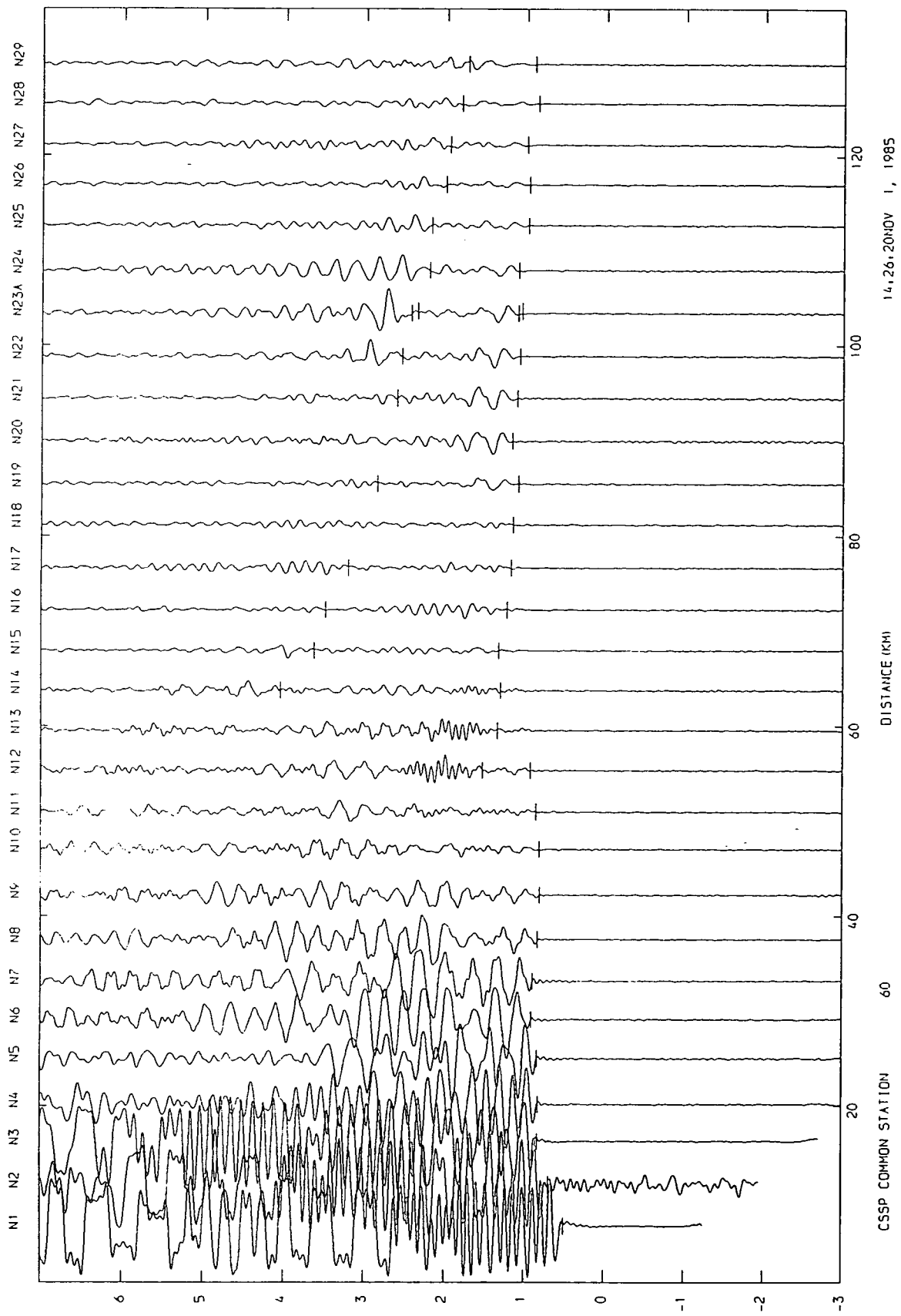
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0023742



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0027502



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0023236



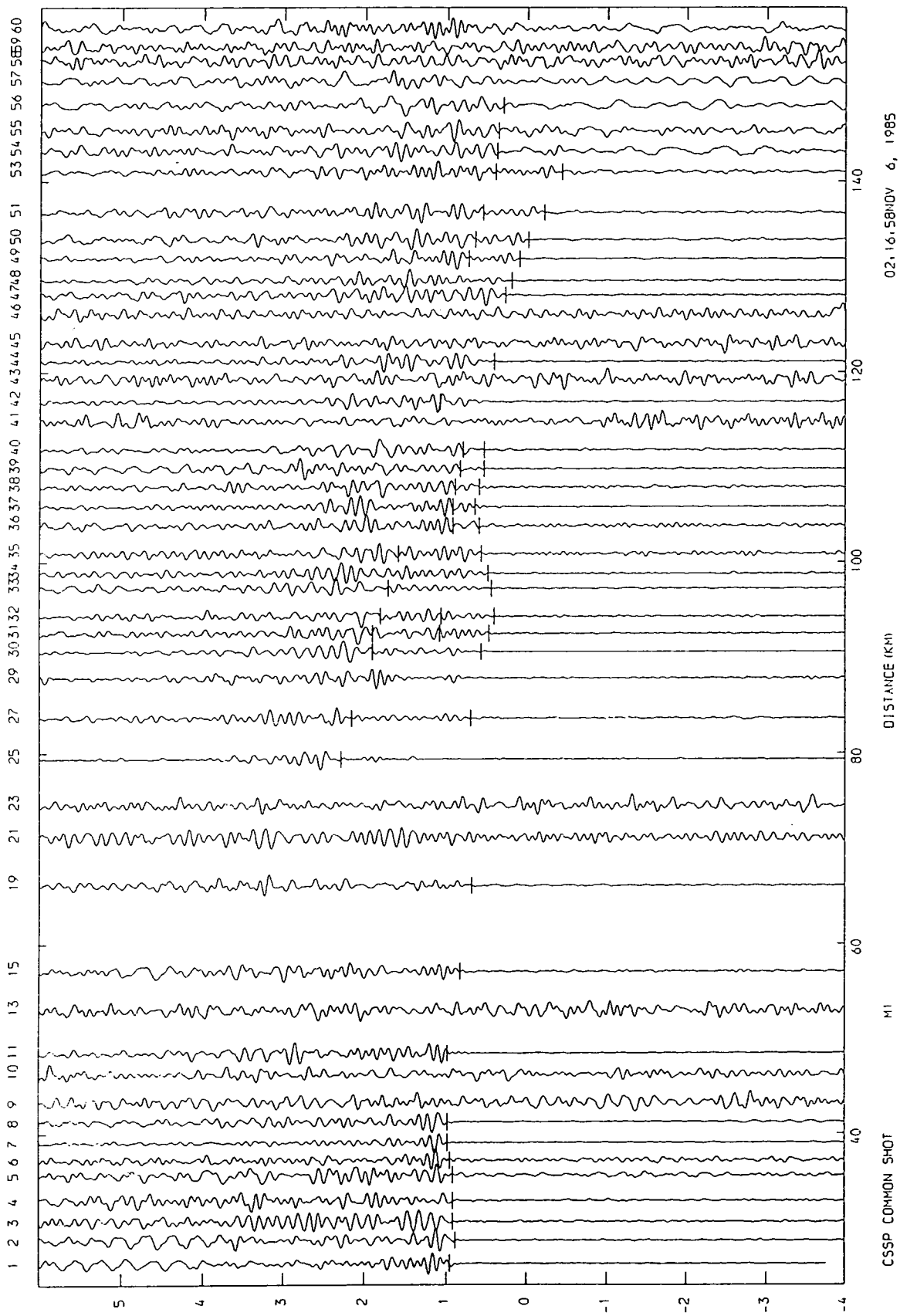
Appendix B

COMMON SHOT SECTION PLOTS WITH TRAVEL-TIME PICKS

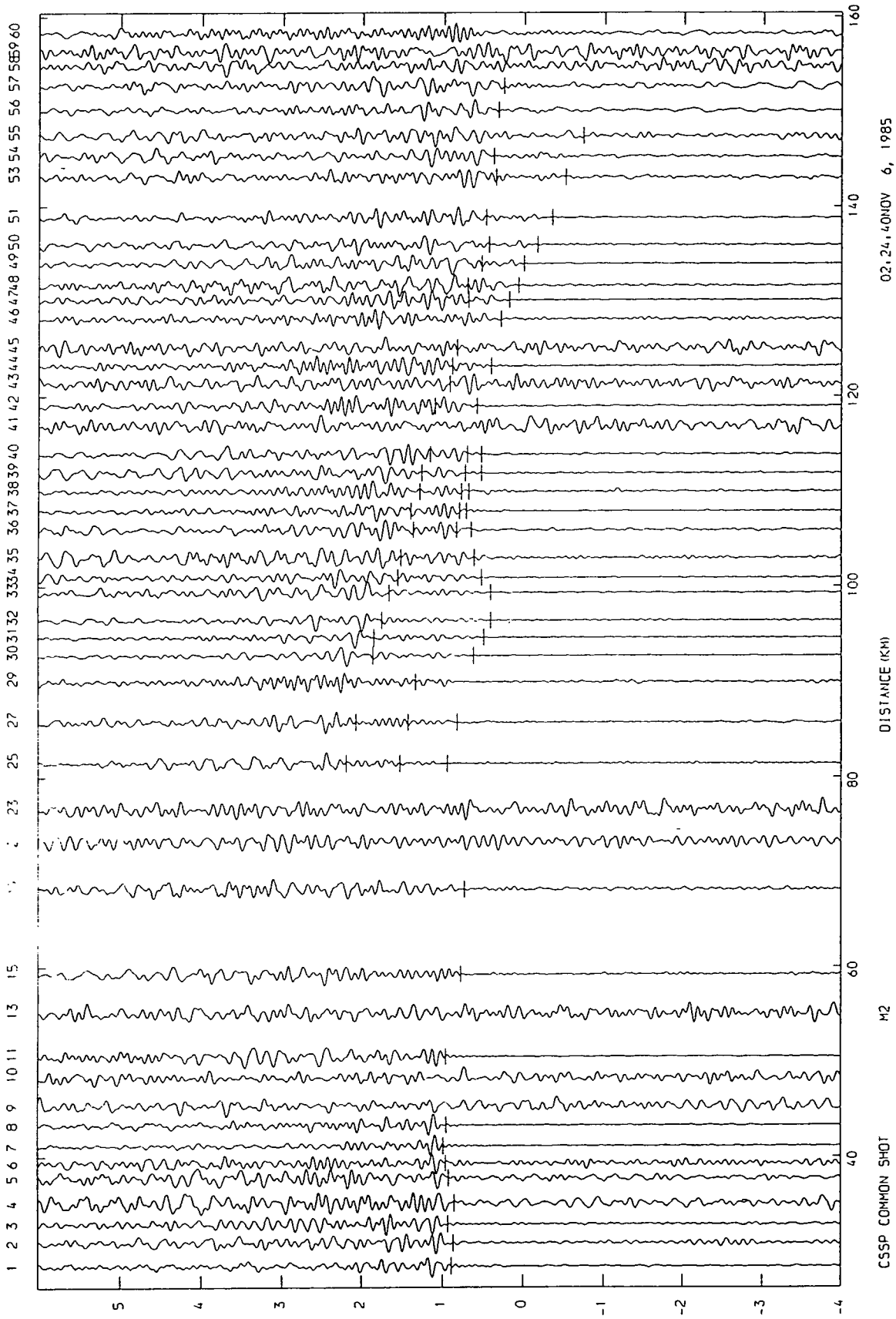
The plots provided in this appendix are all labelled with the plotting parametres and the name of the section plotted. In summary they are:

1. Common shot sections for the Irish Sea shots with equalized amplitudes
2. Common shot sections for the North Sea shots with equalized amplitudes

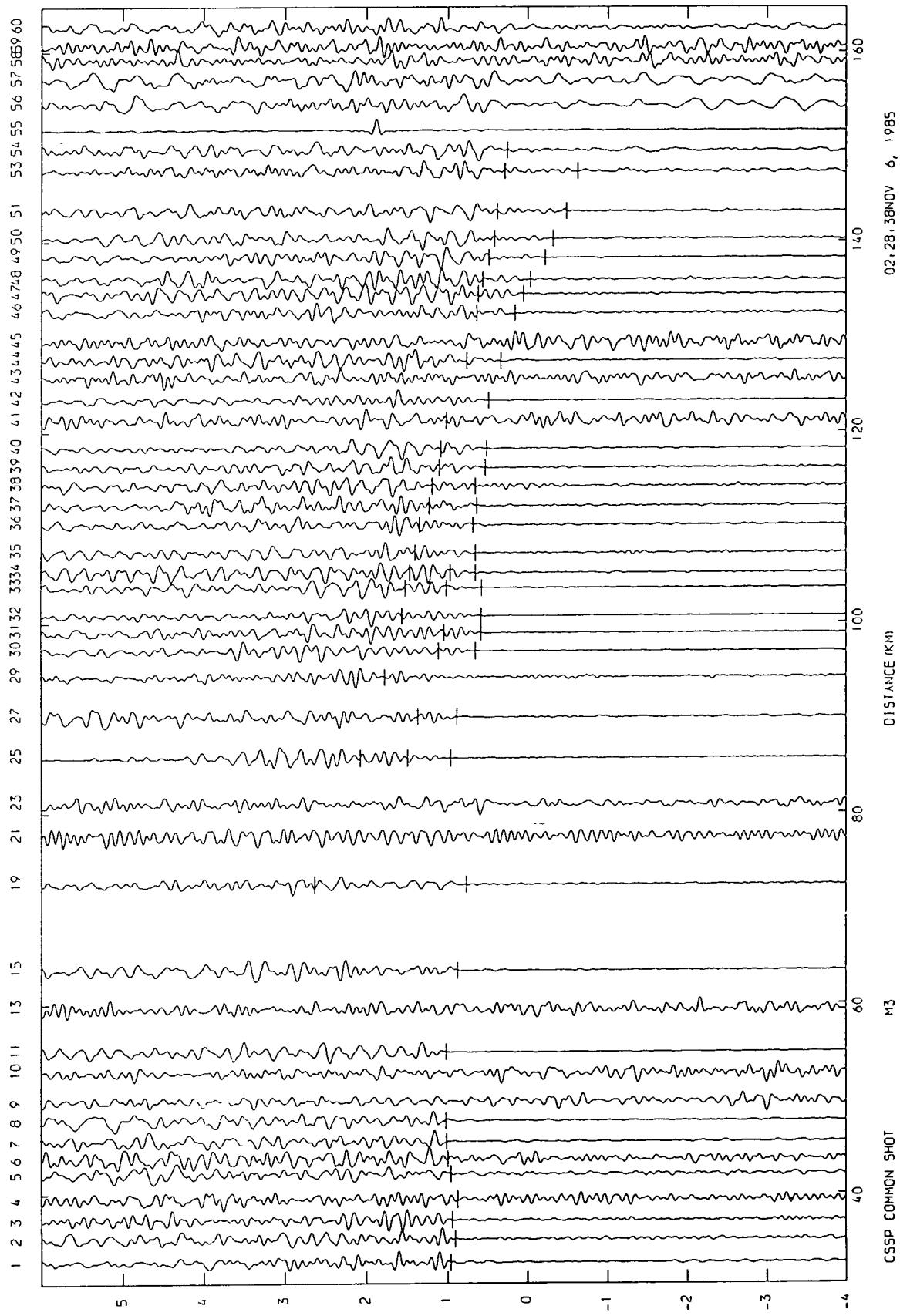
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



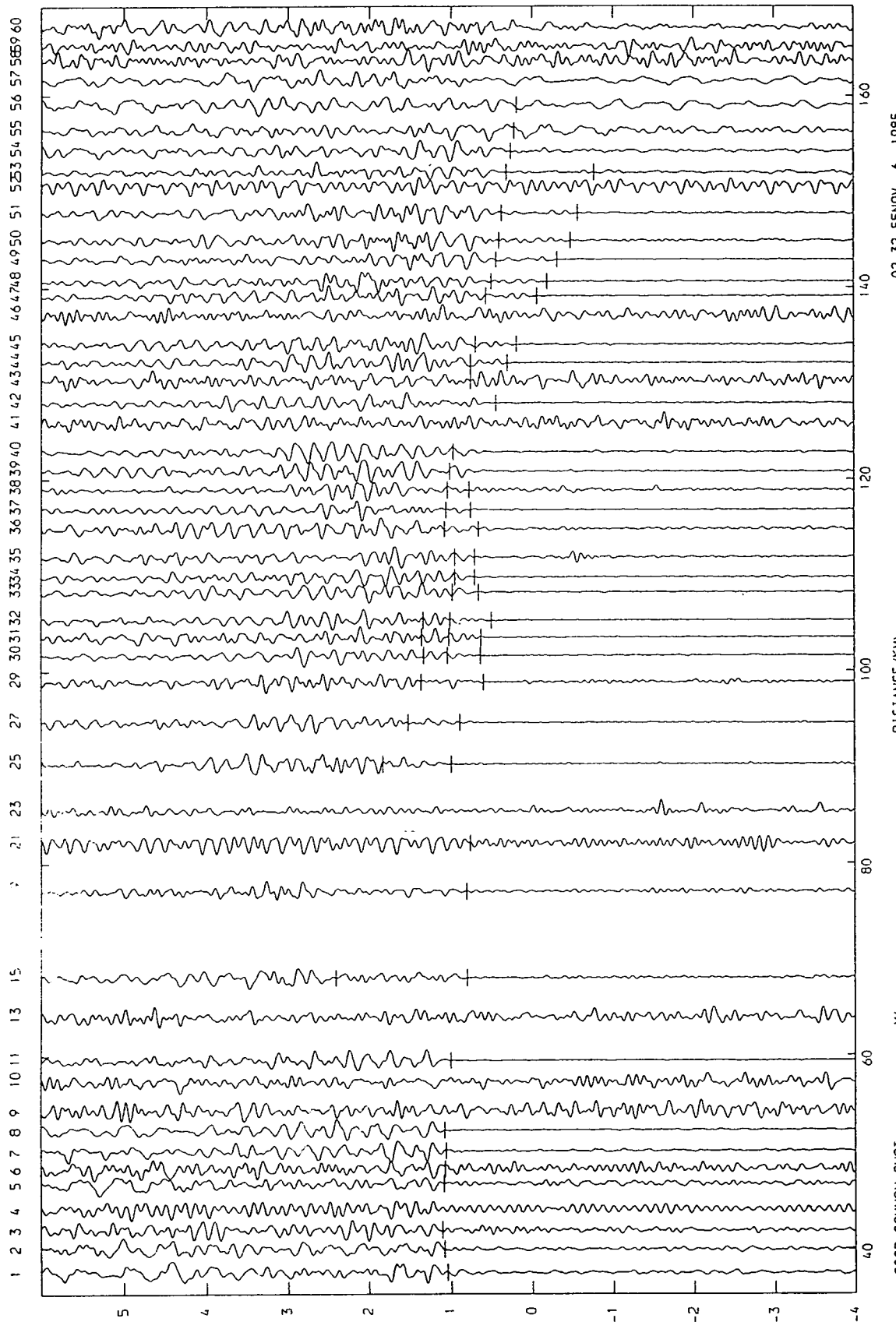
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



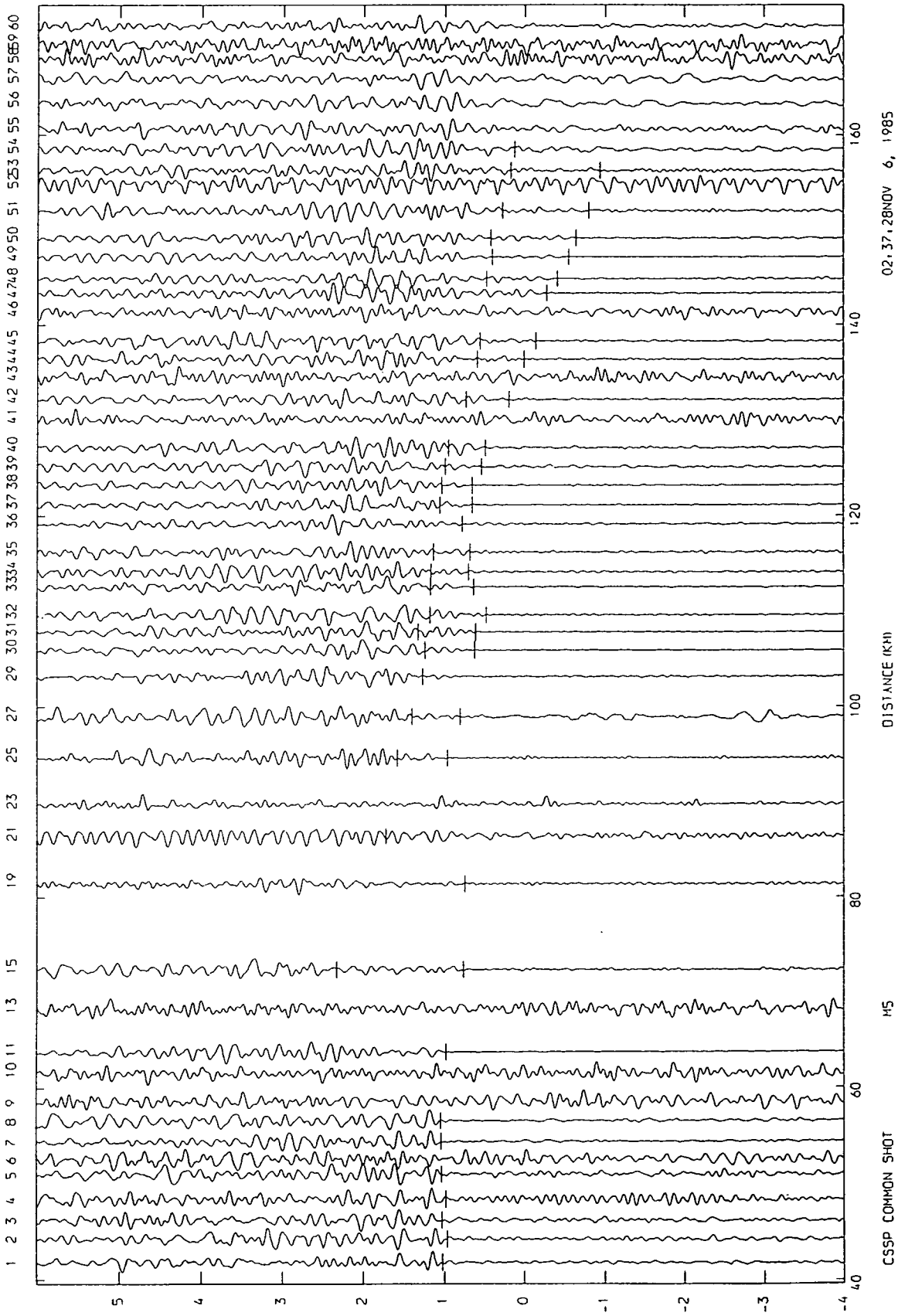
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



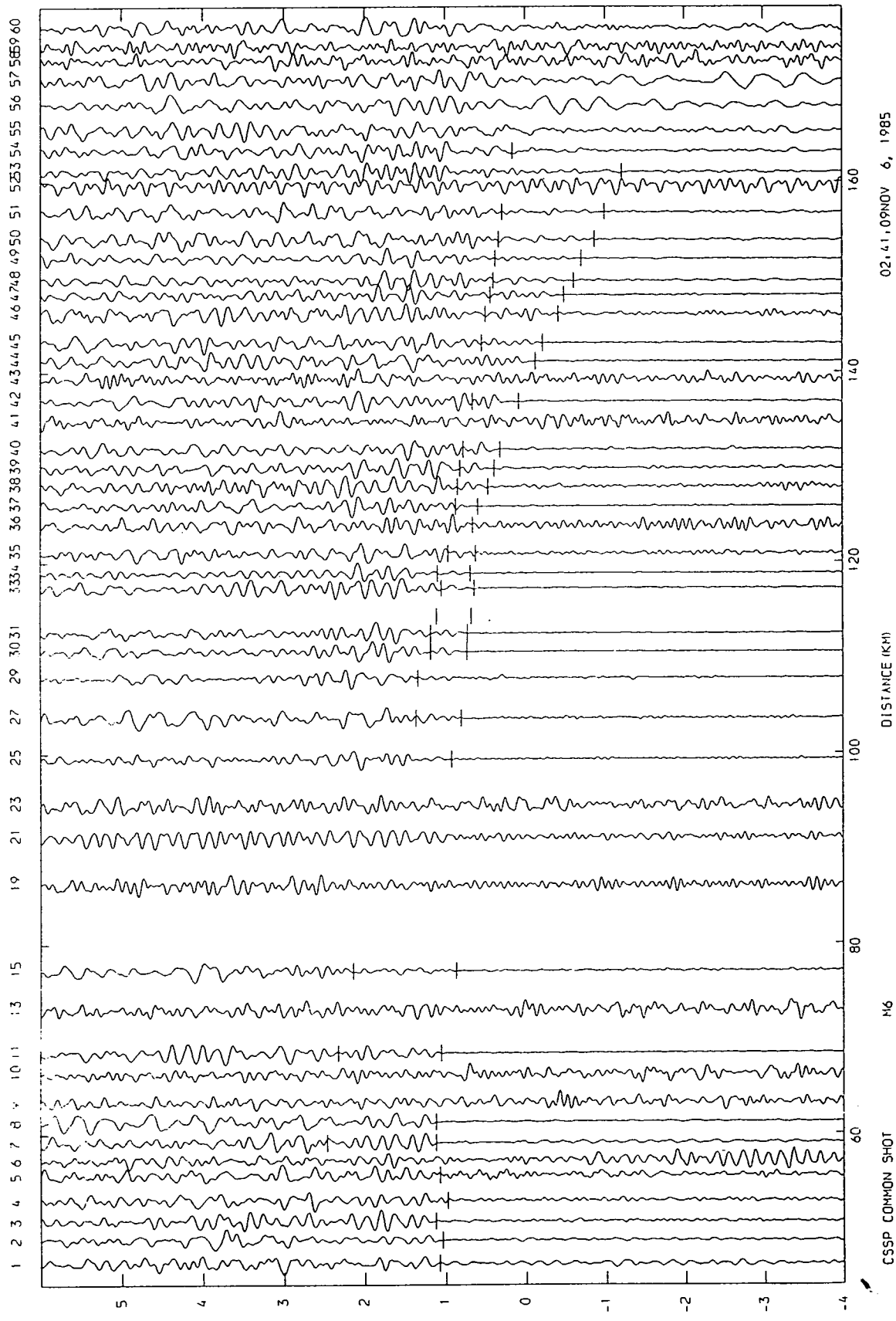
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



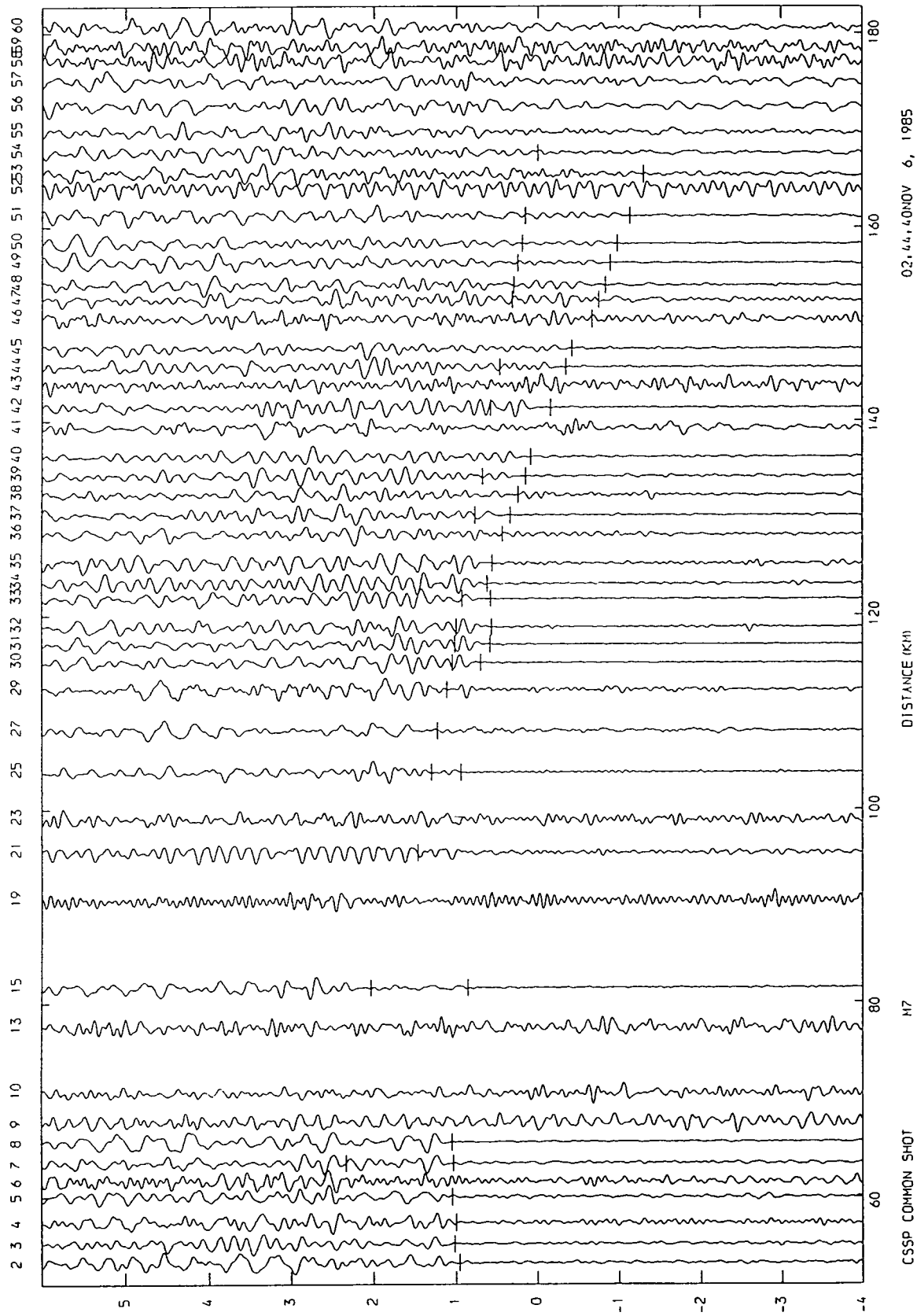
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



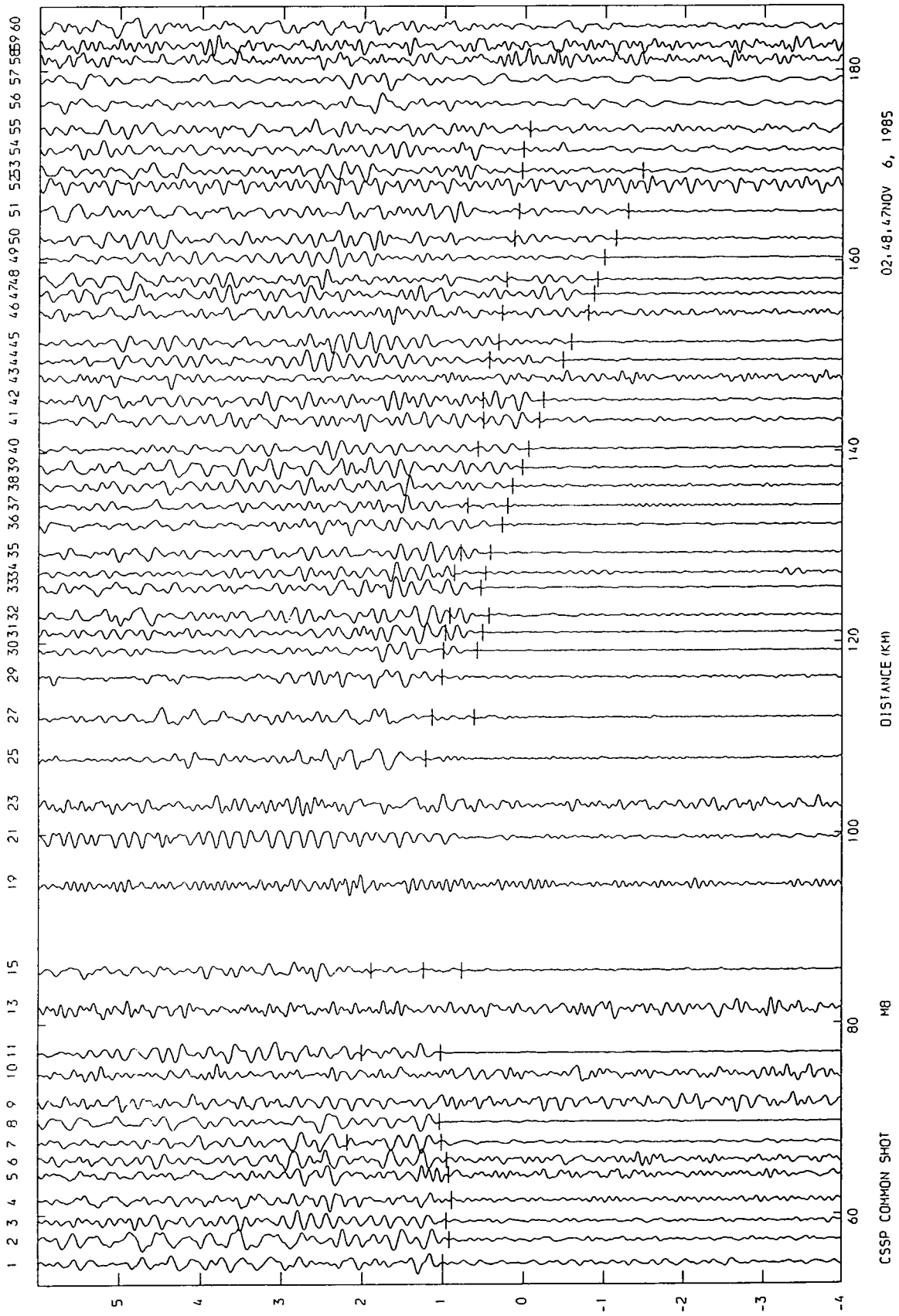
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



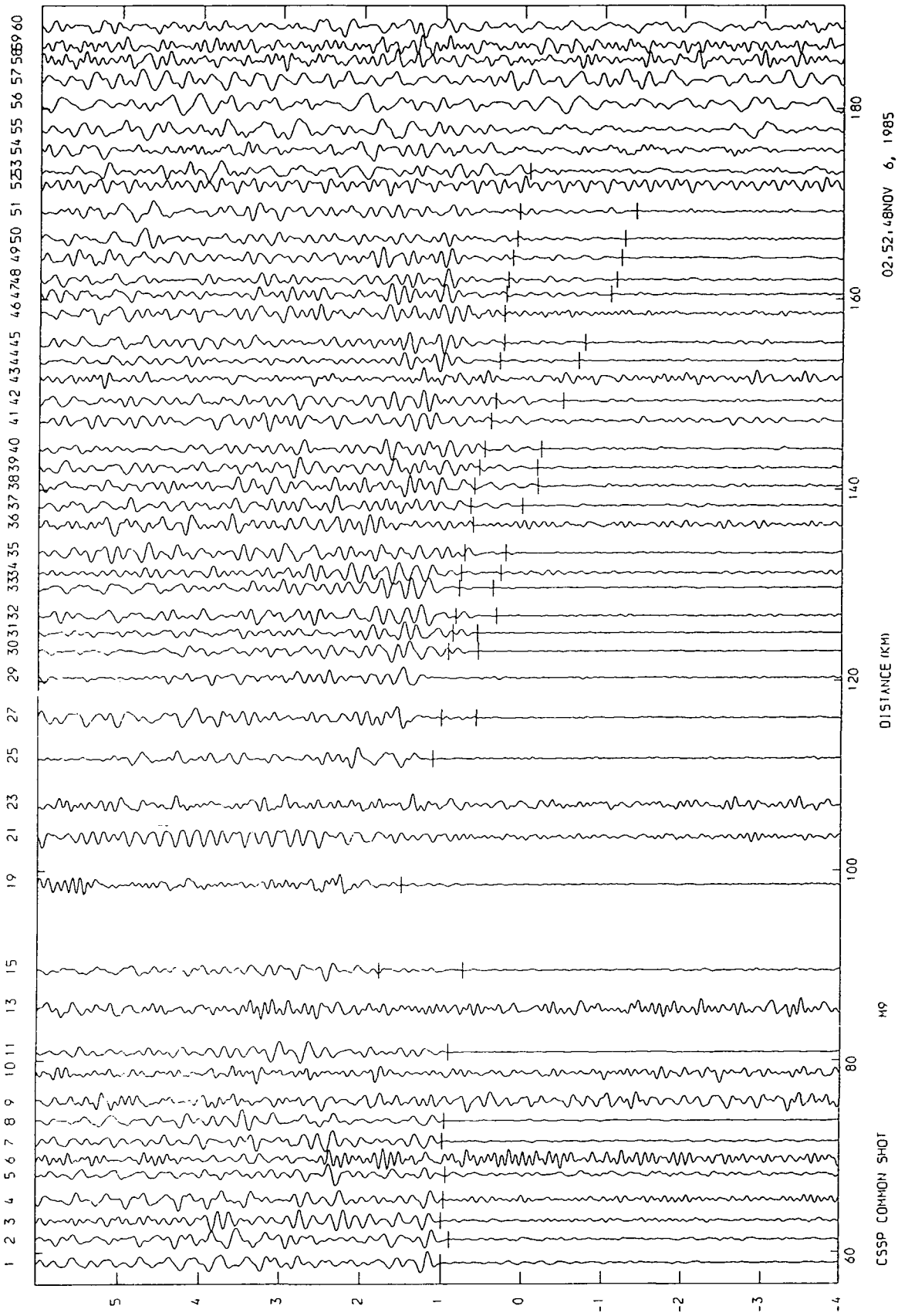
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



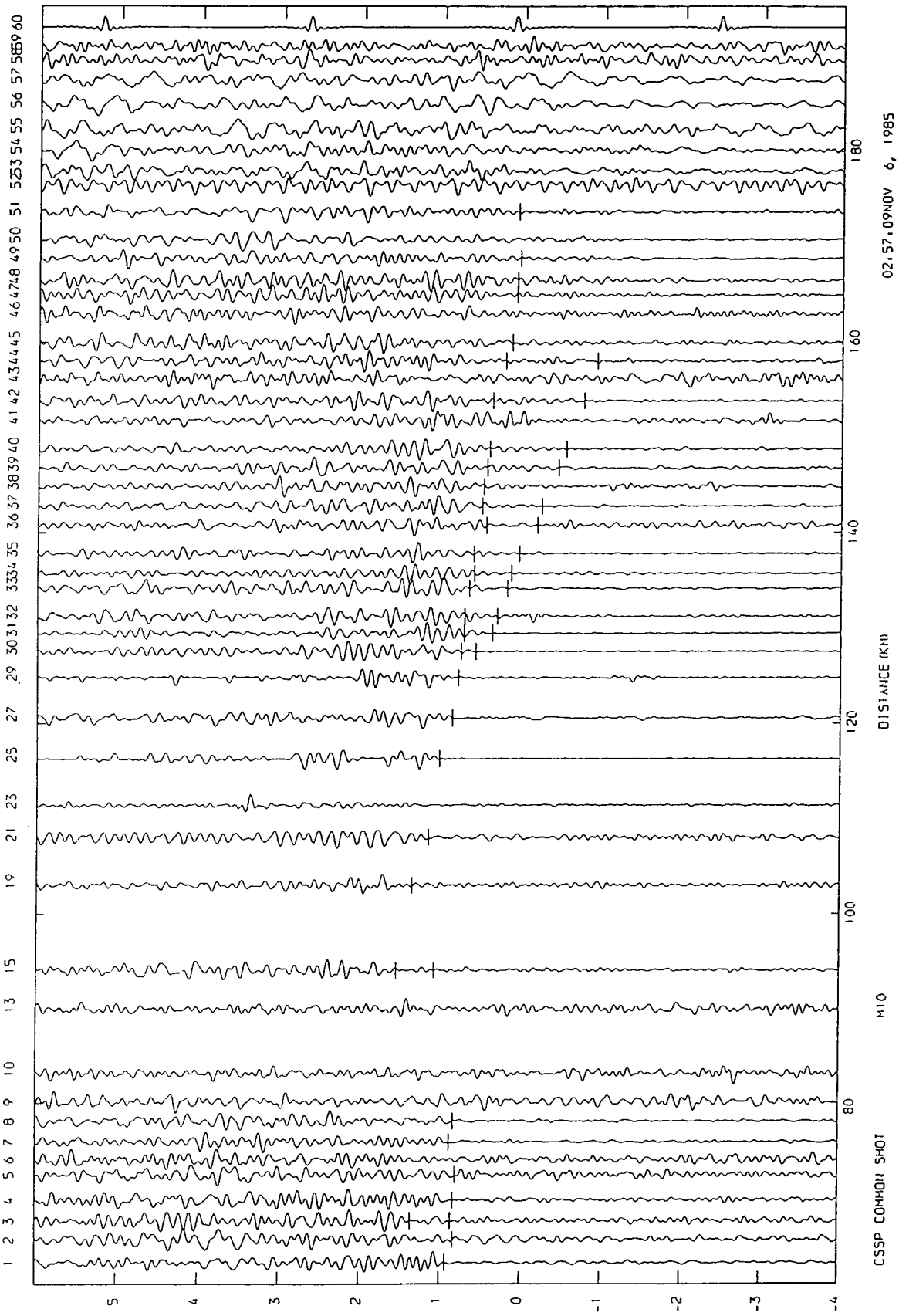
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



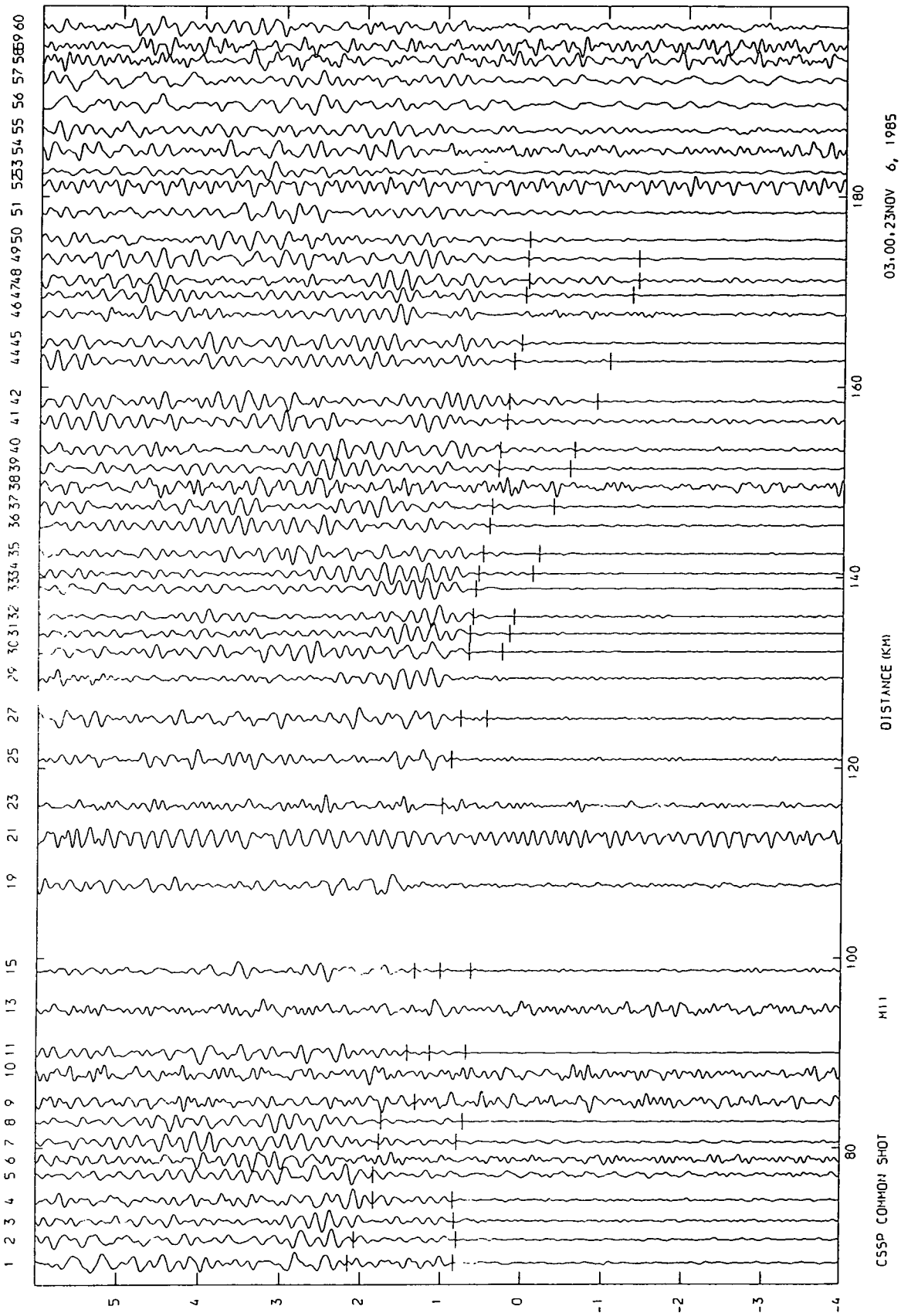
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



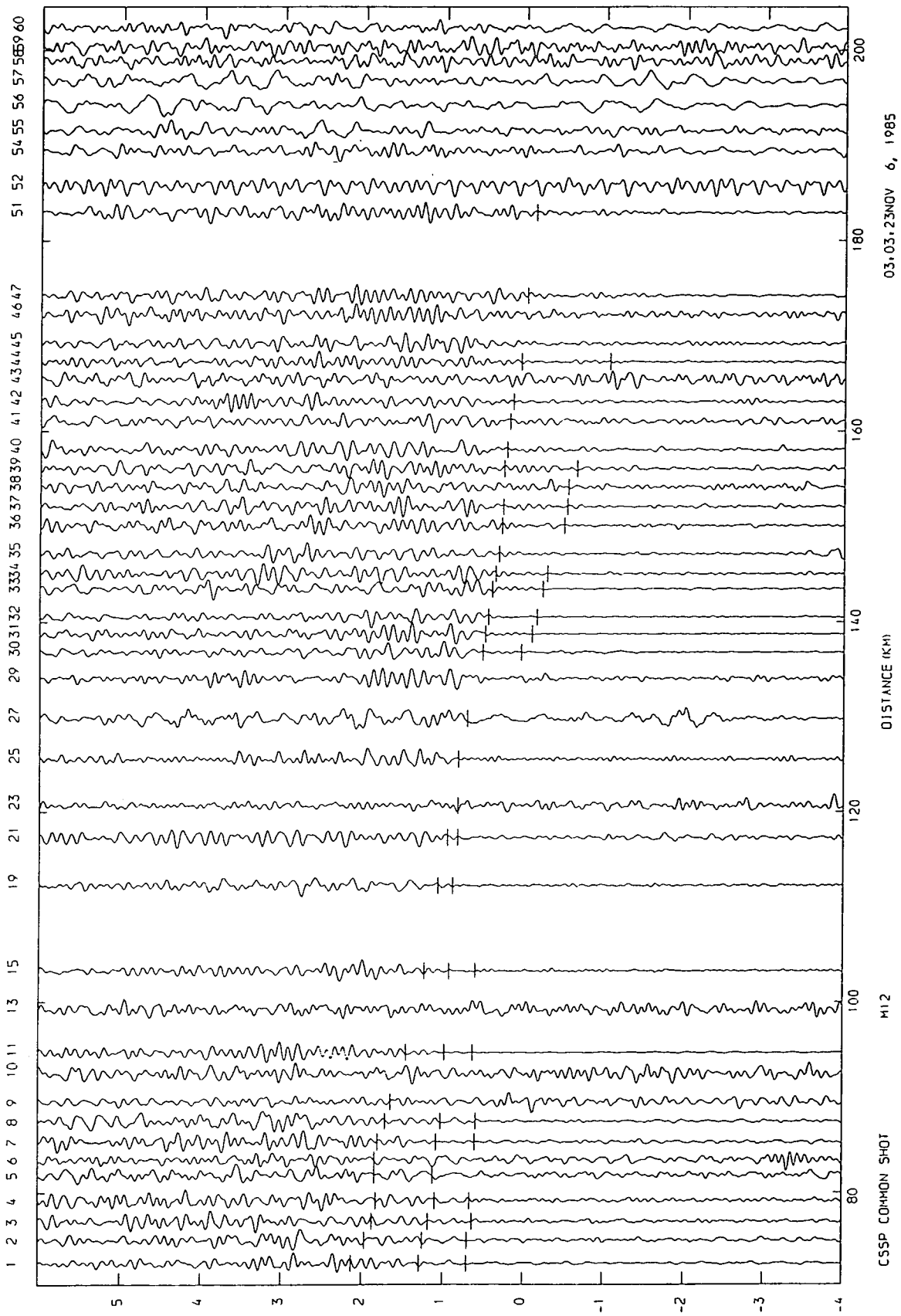
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



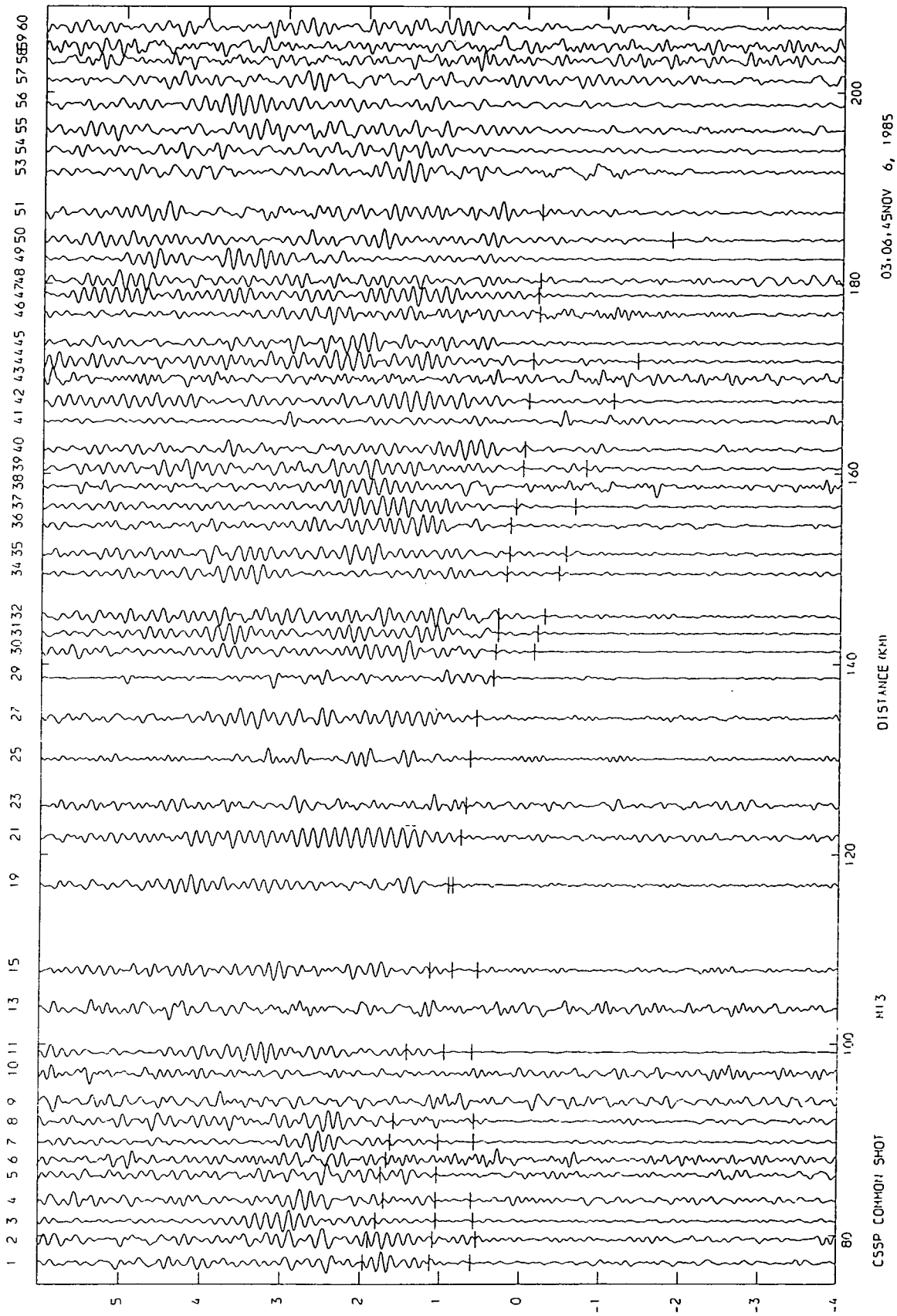
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



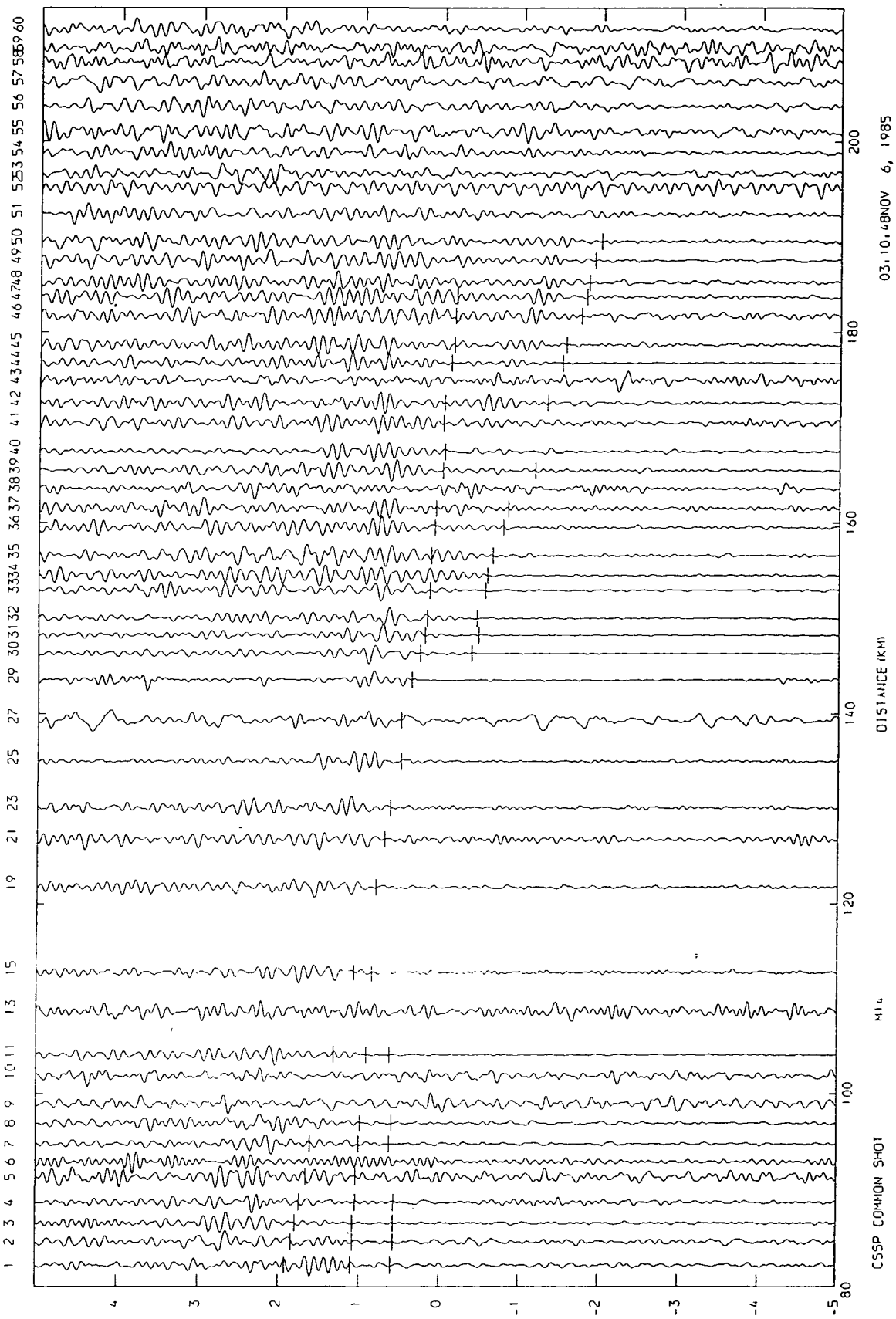
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



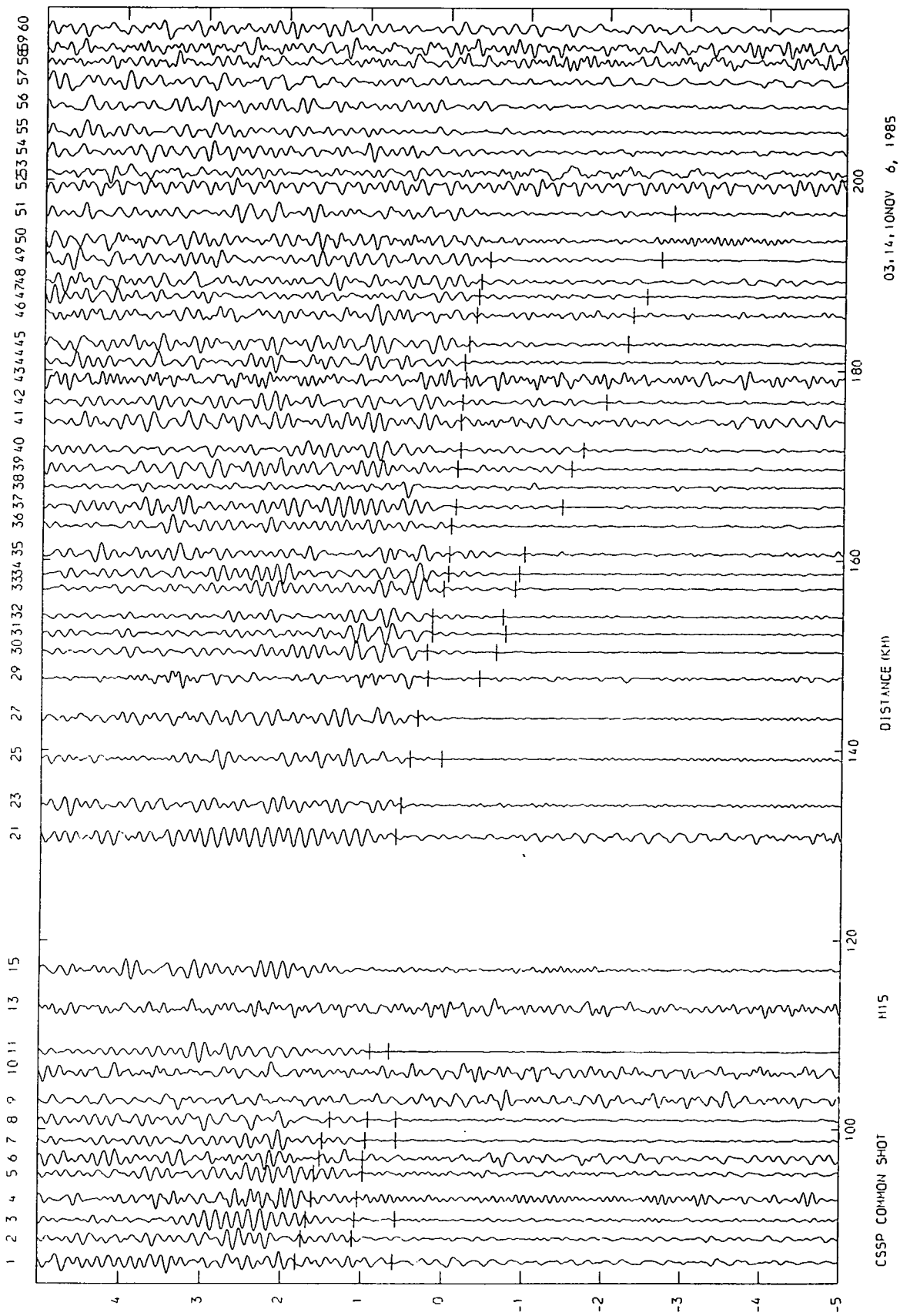
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



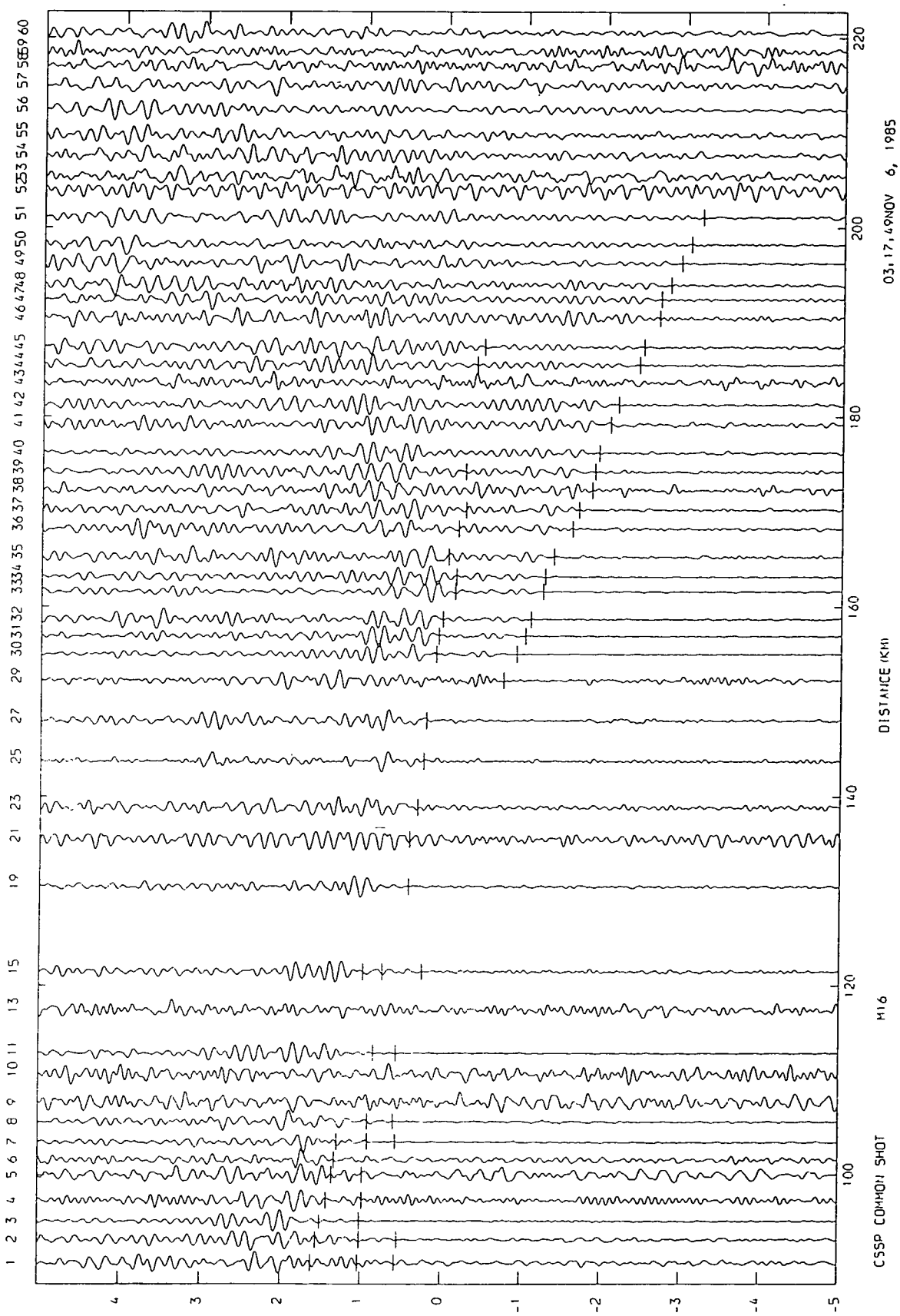
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



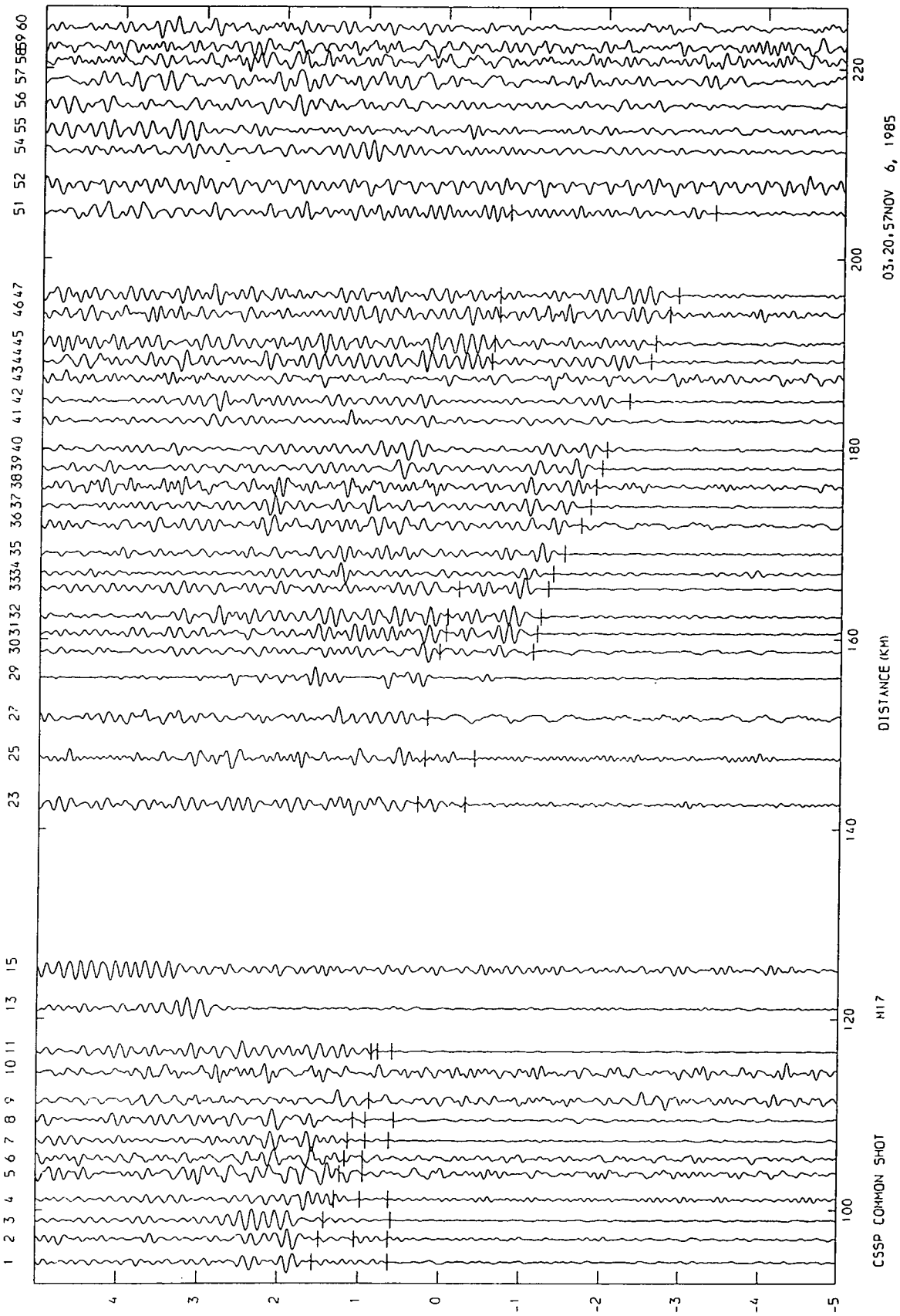
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



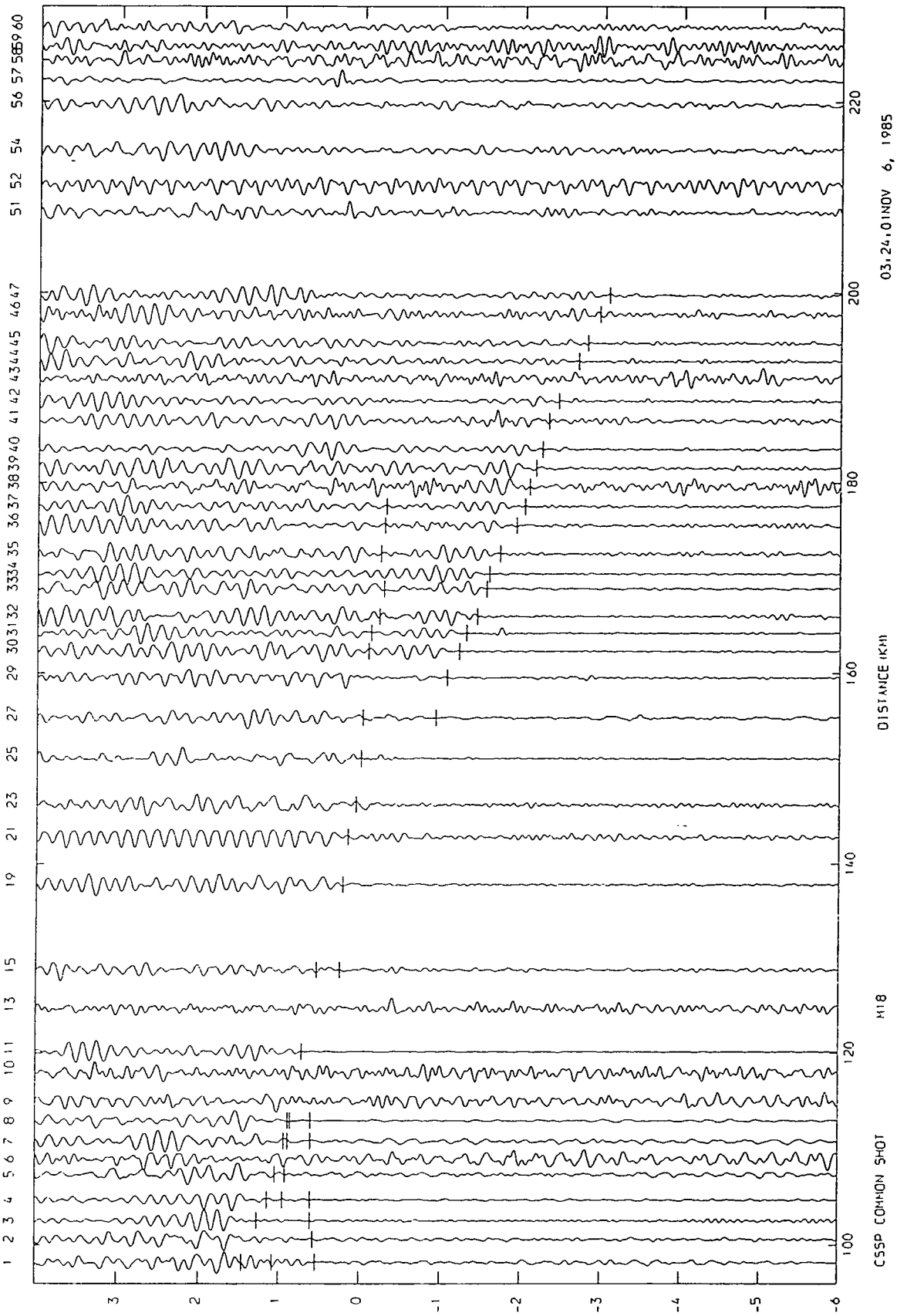
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



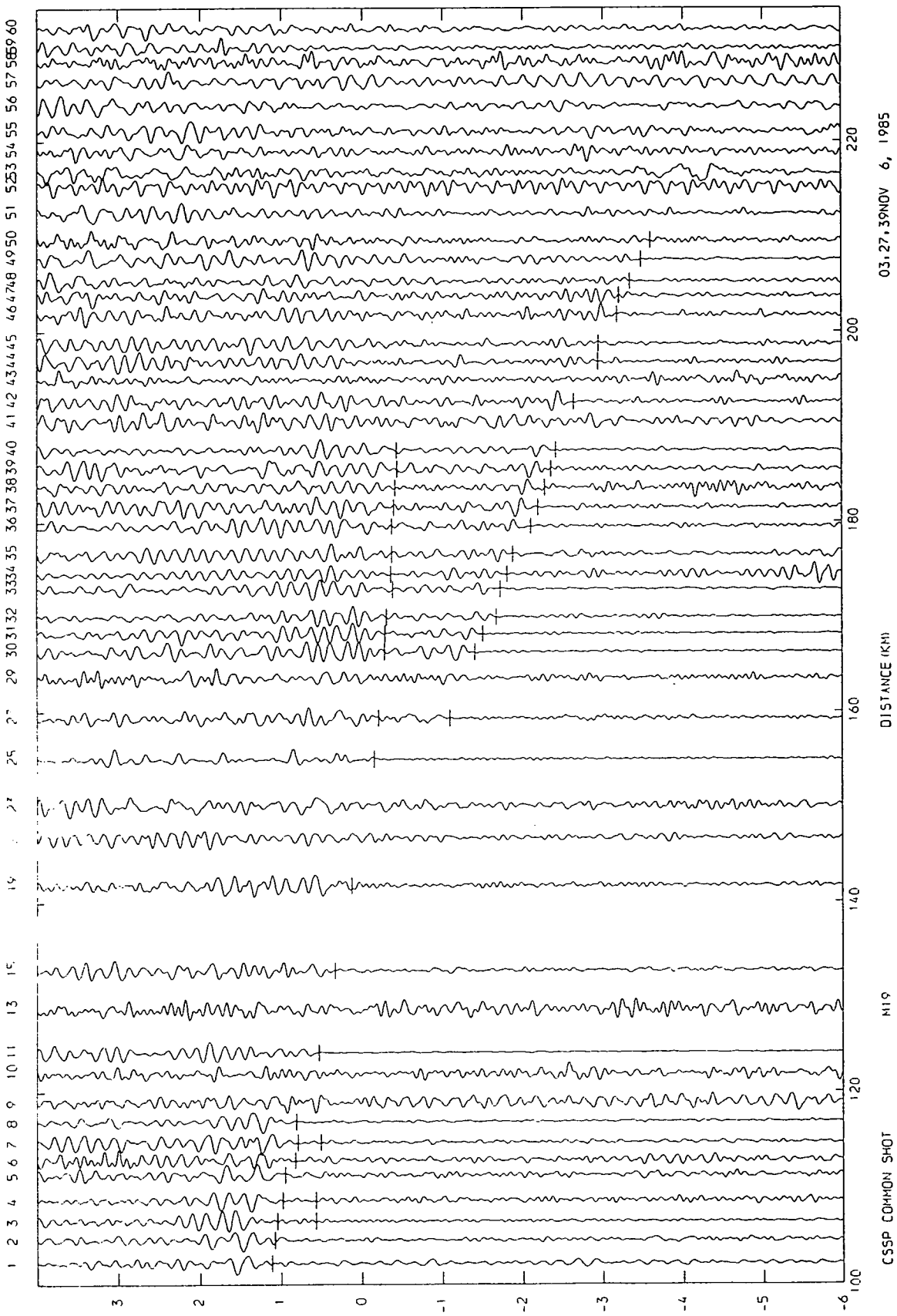
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



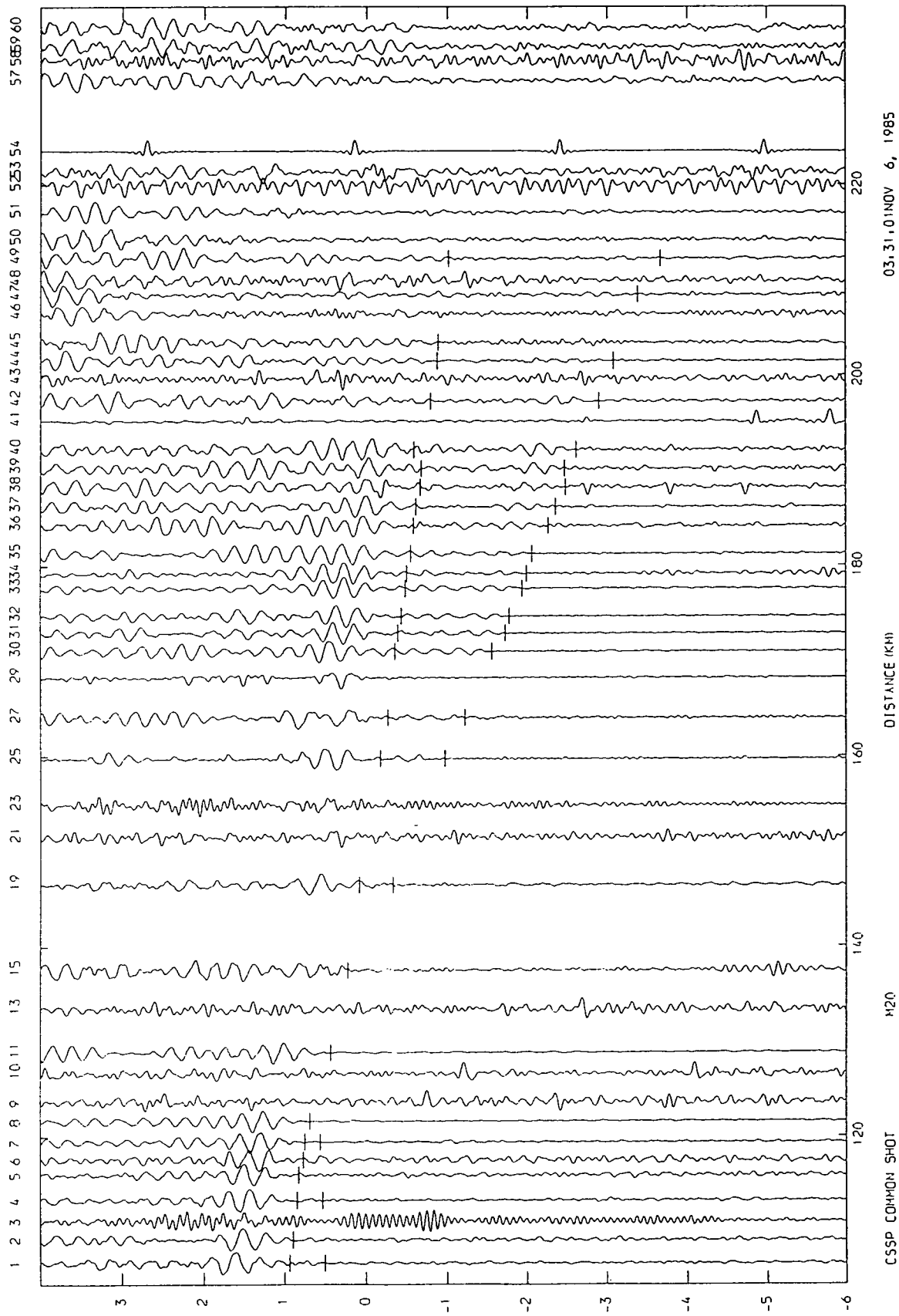
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



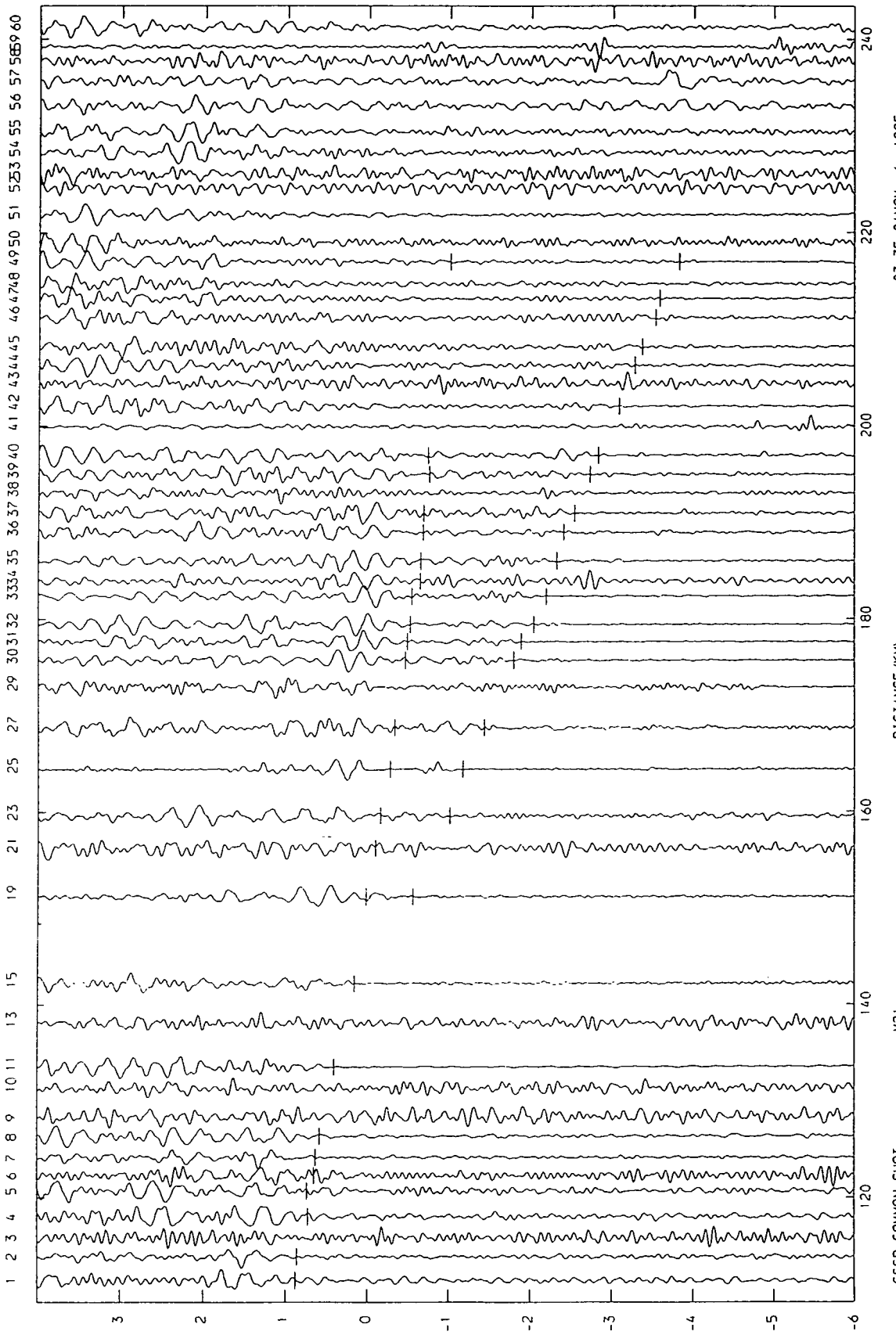
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



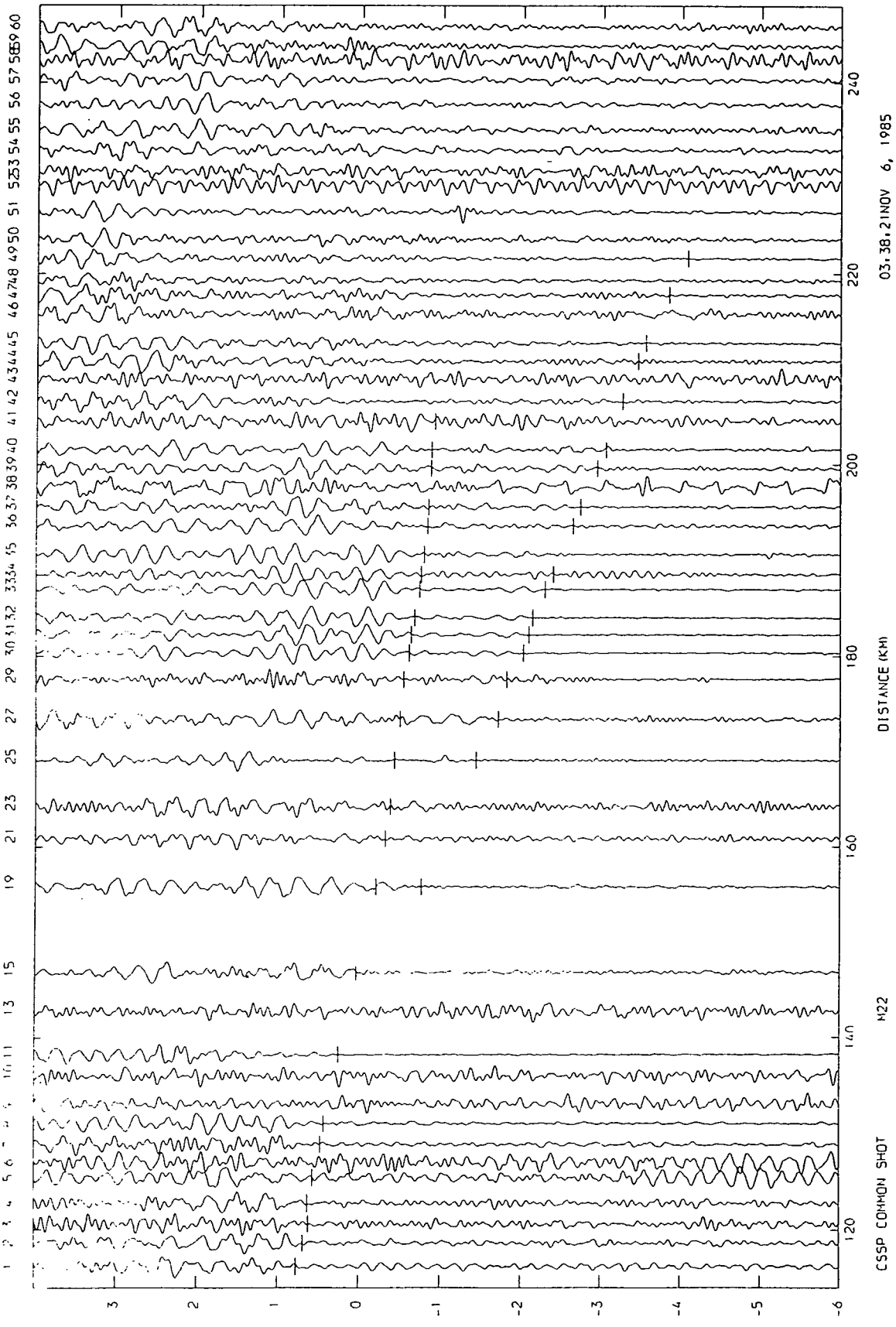
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



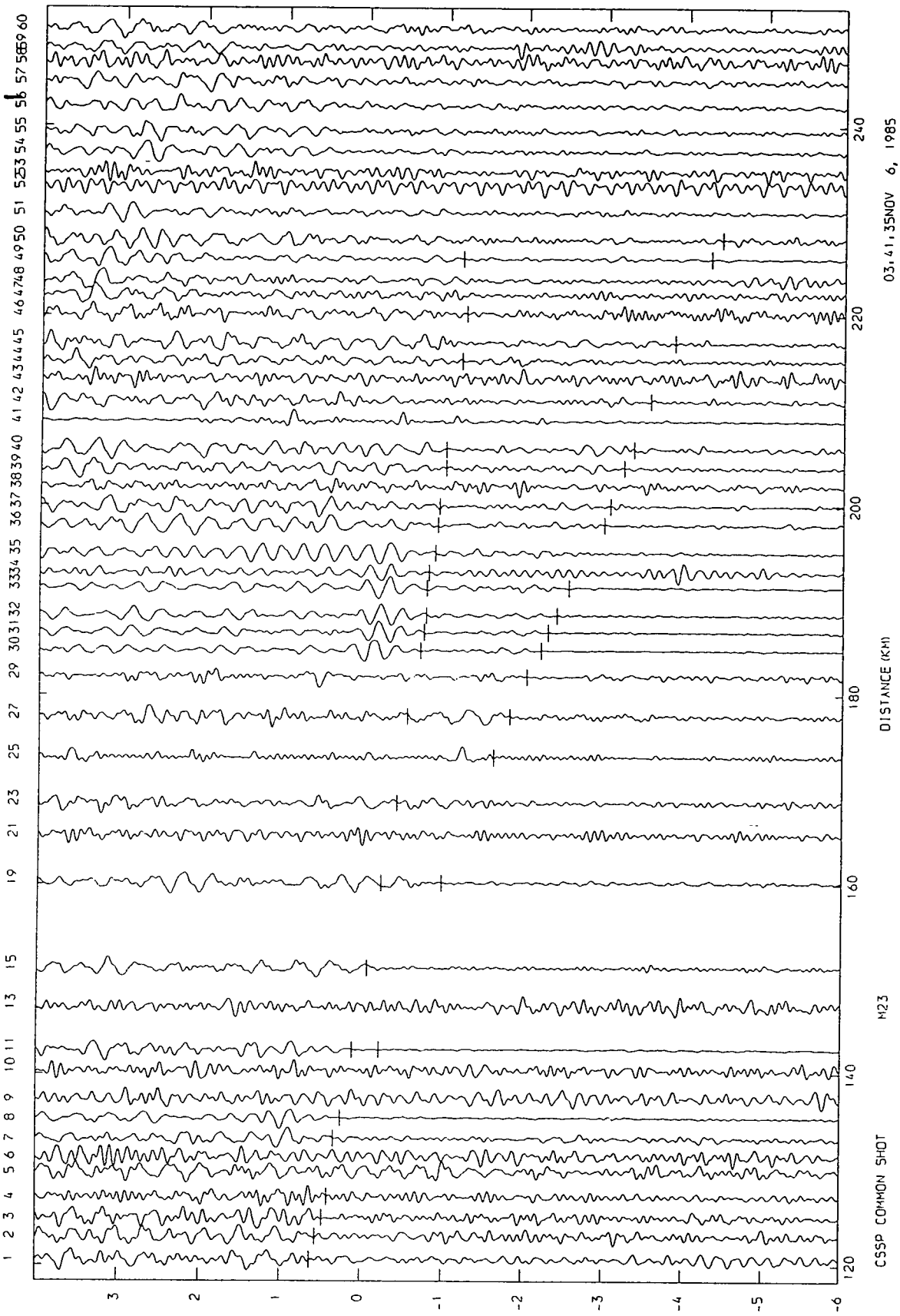
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



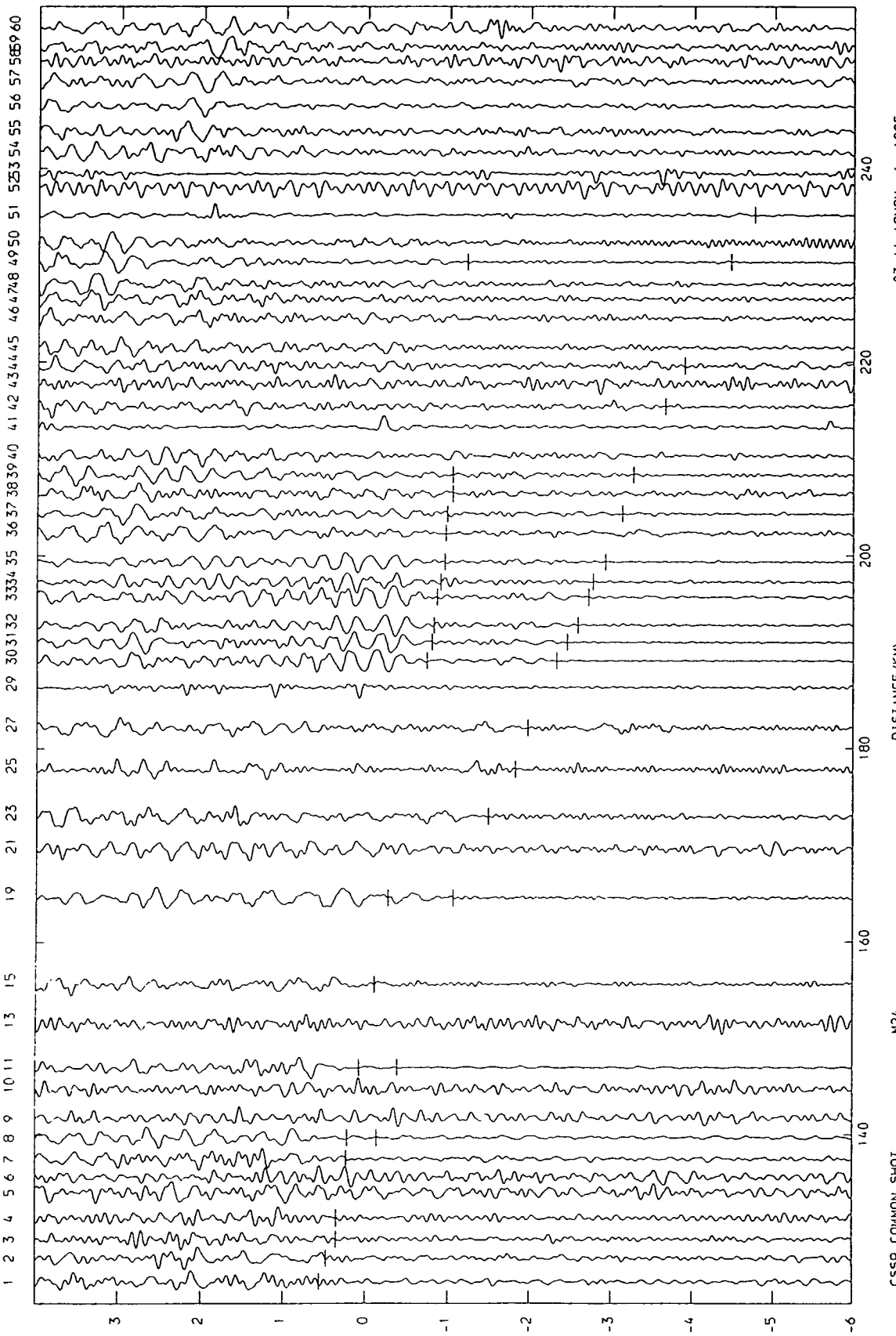
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



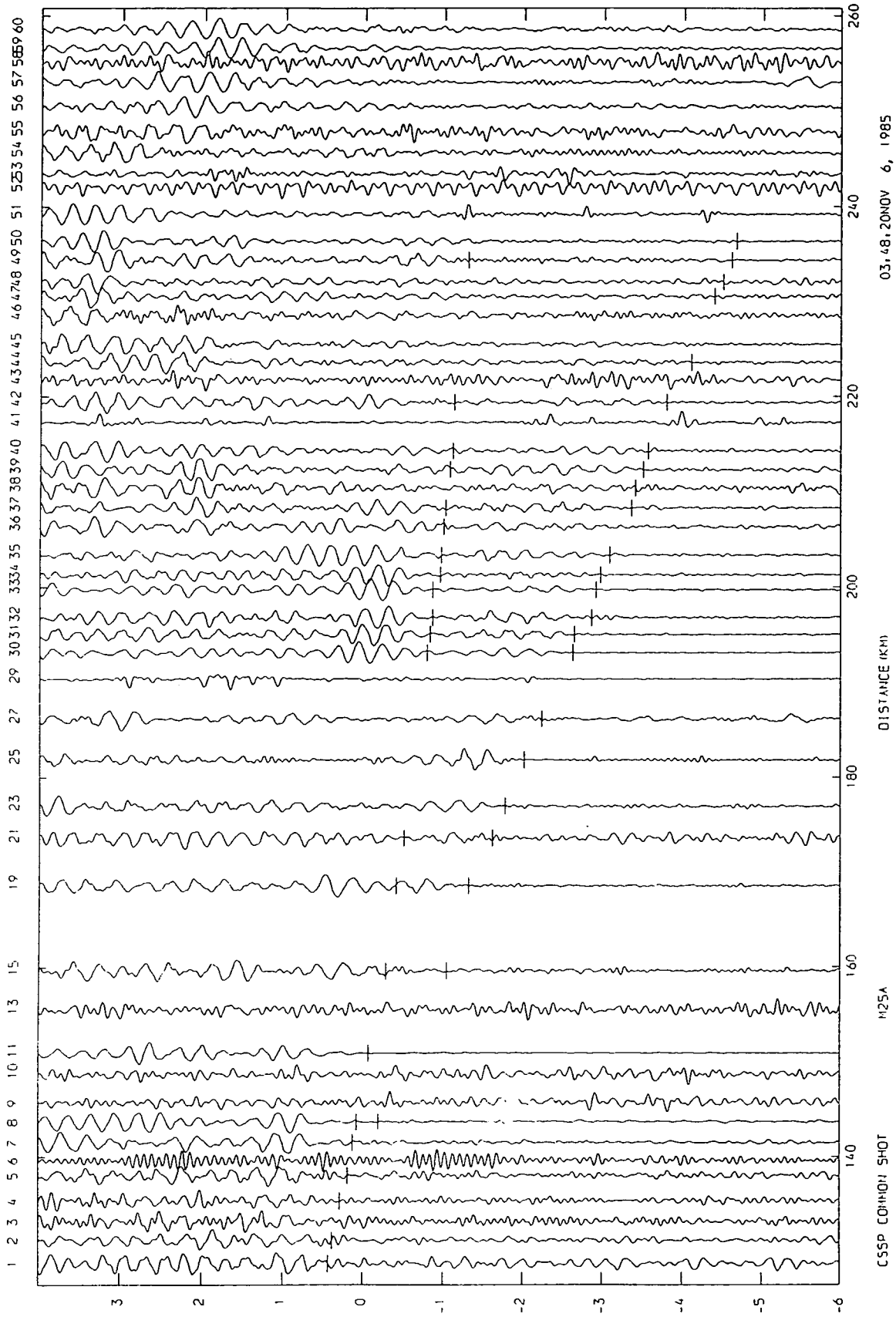
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



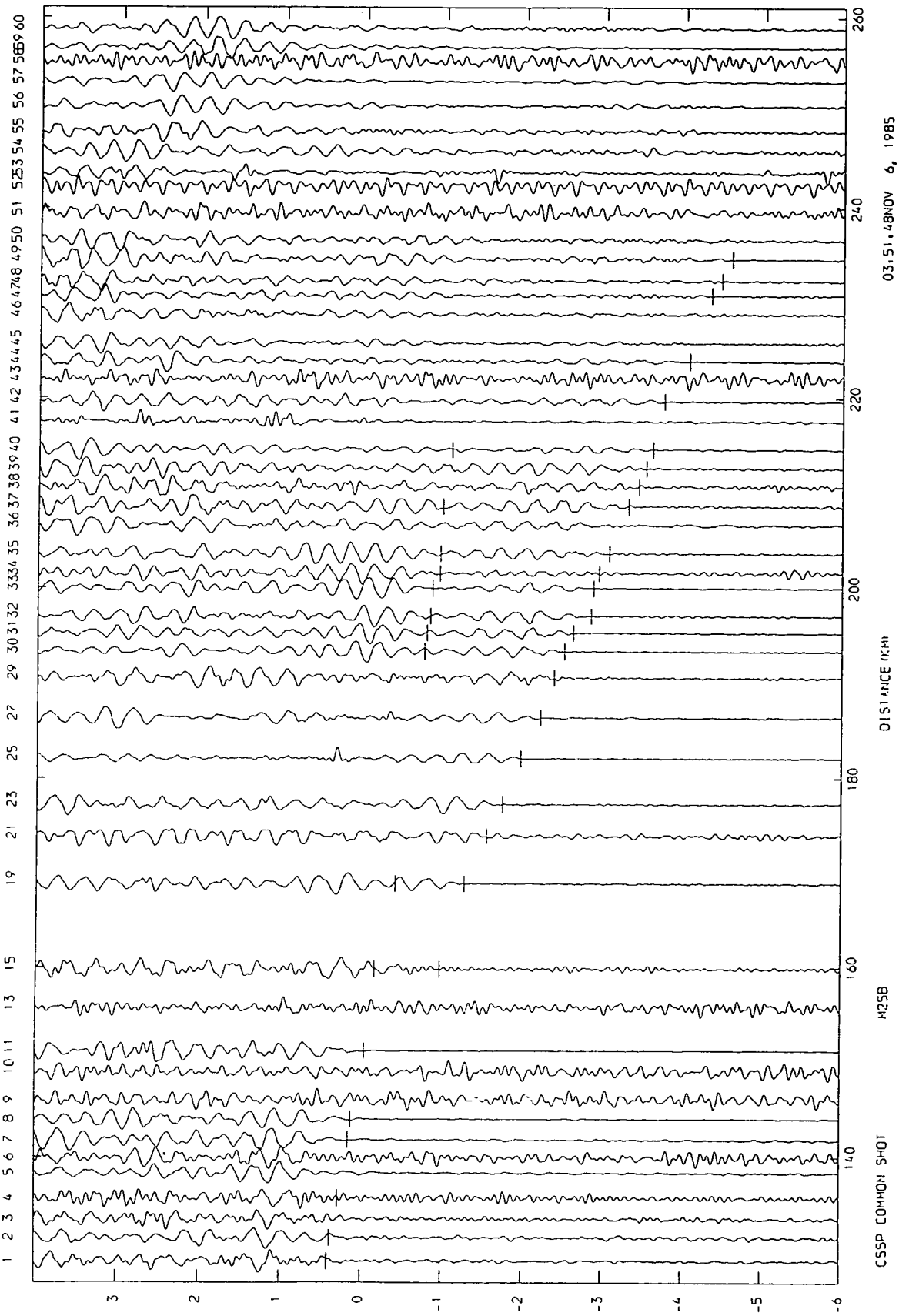
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



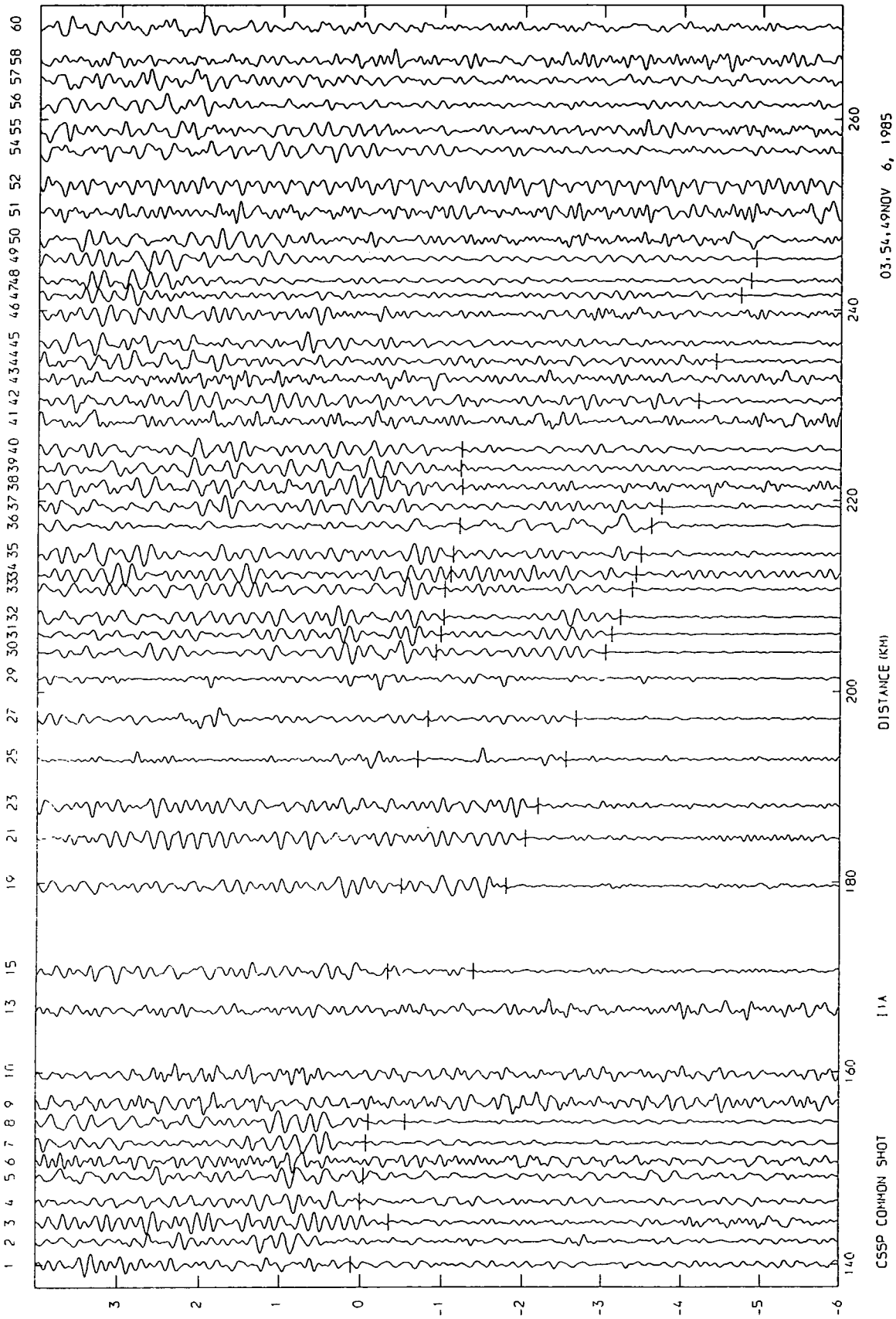
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



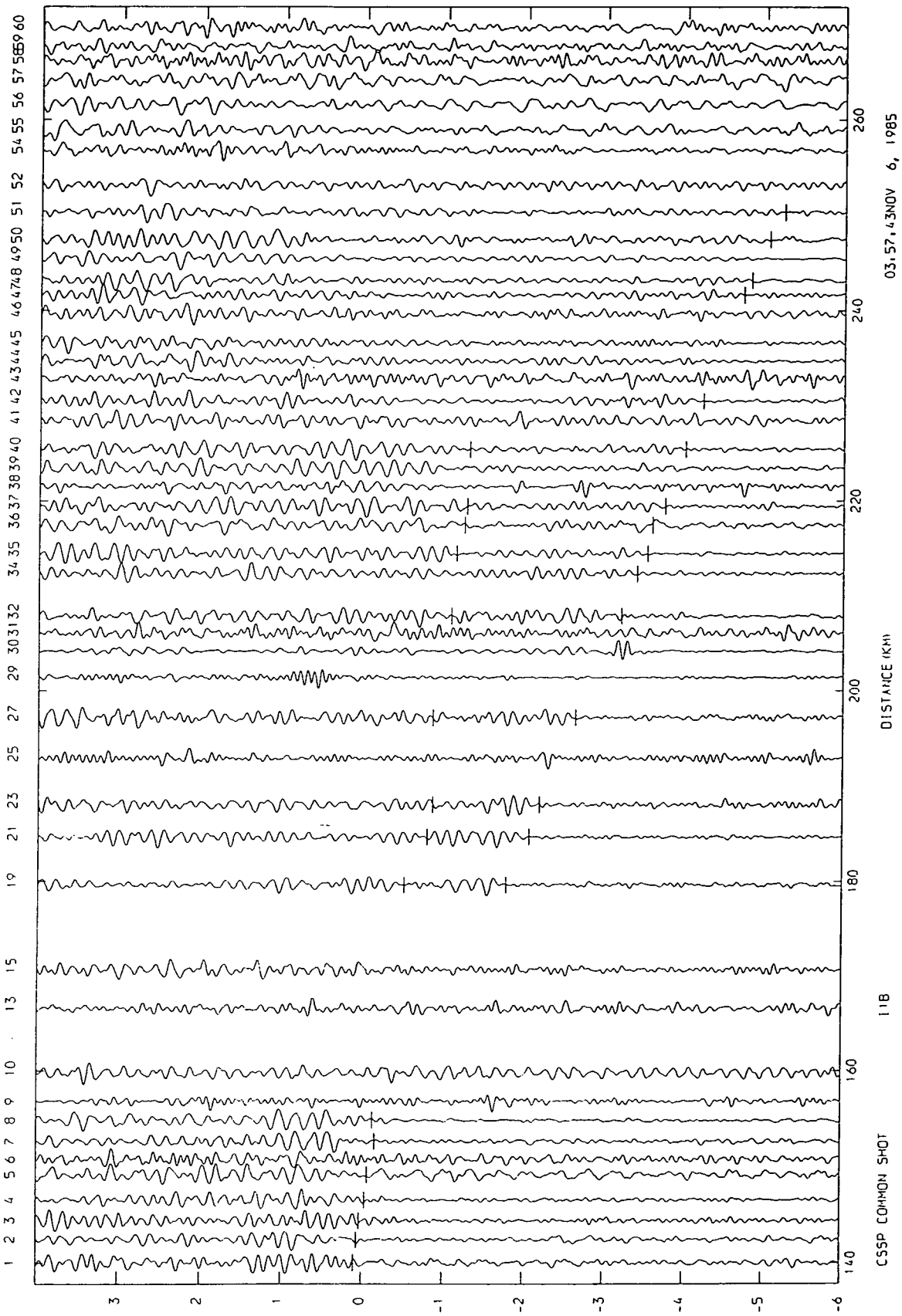
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



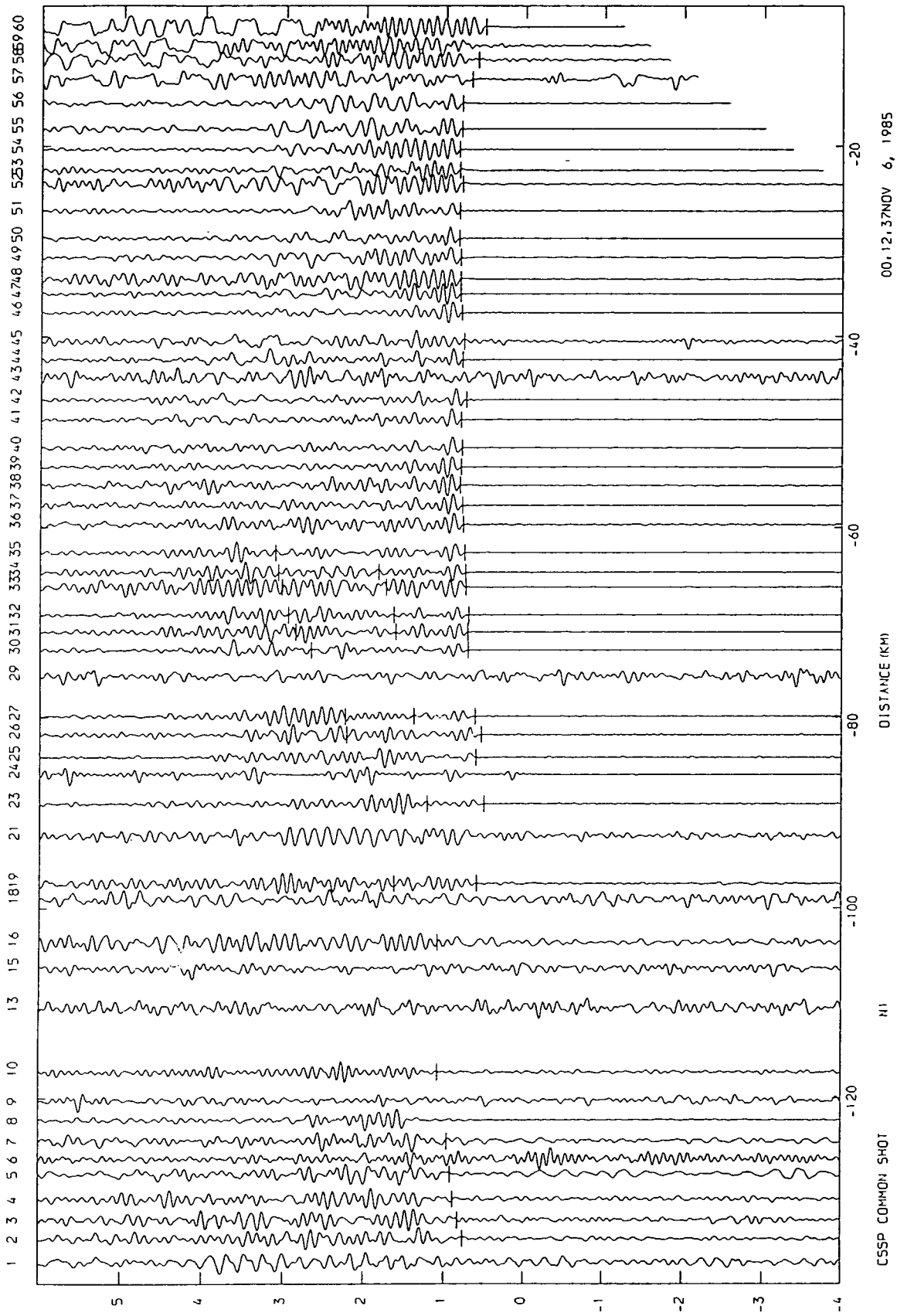
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



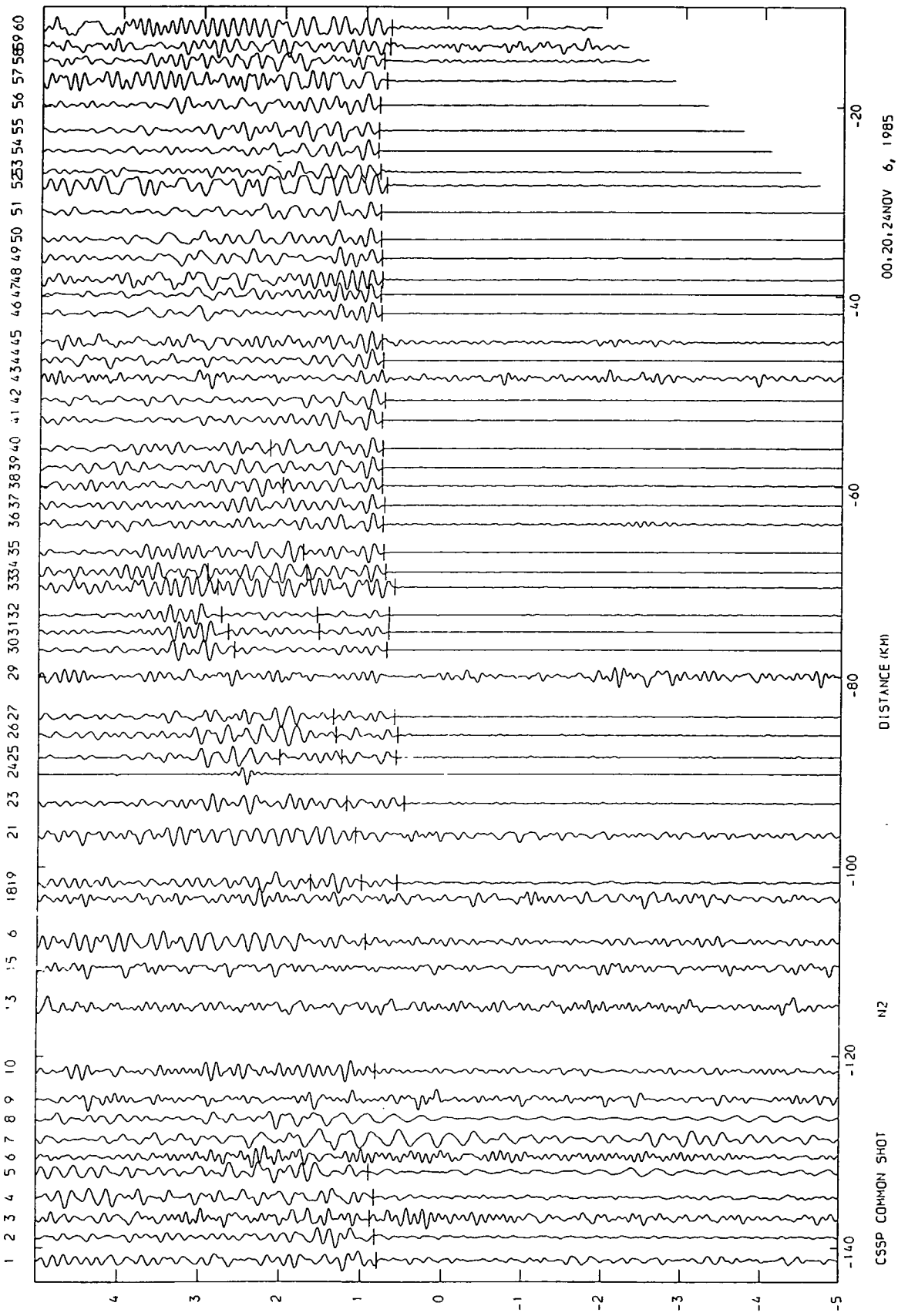
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



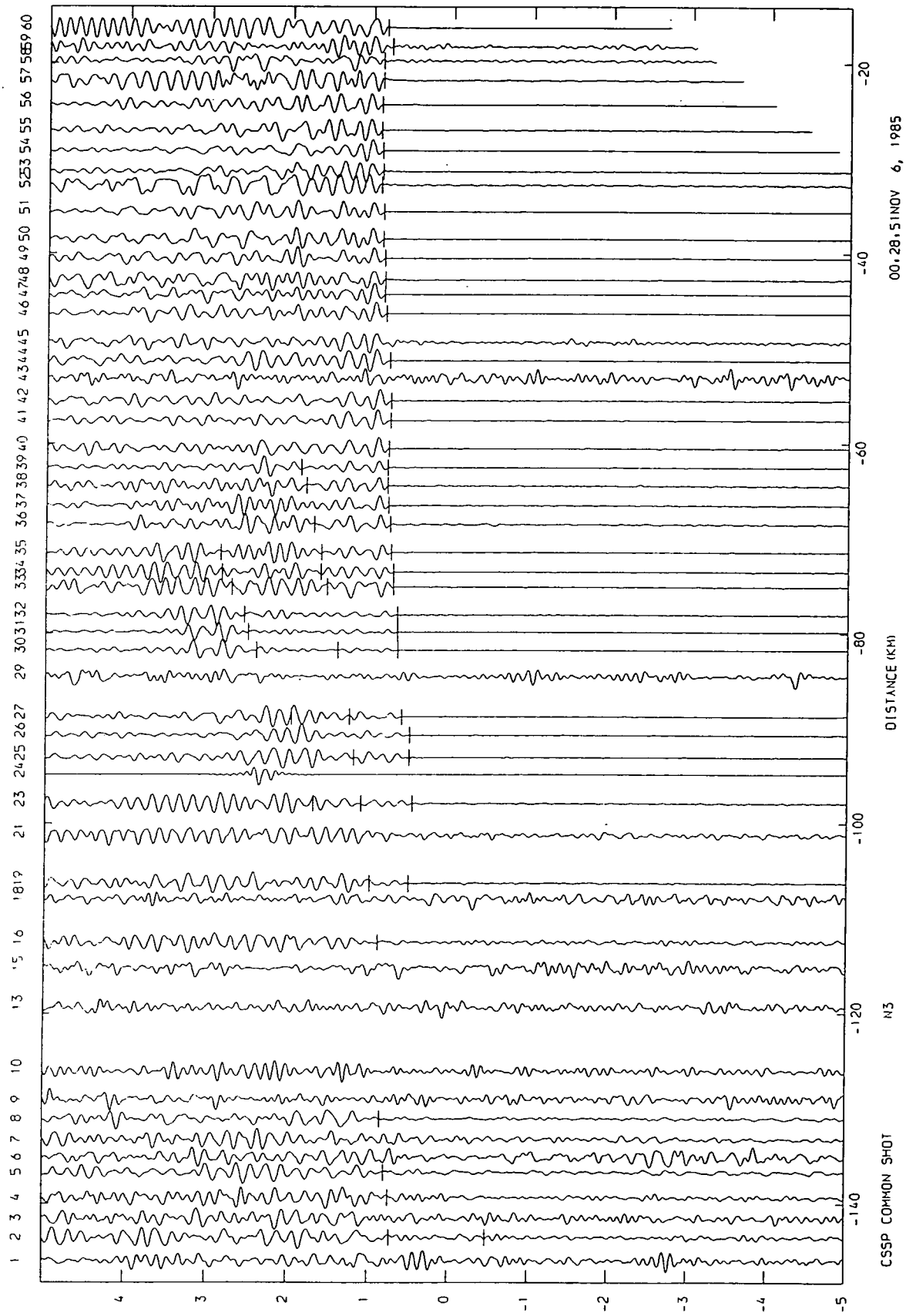
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



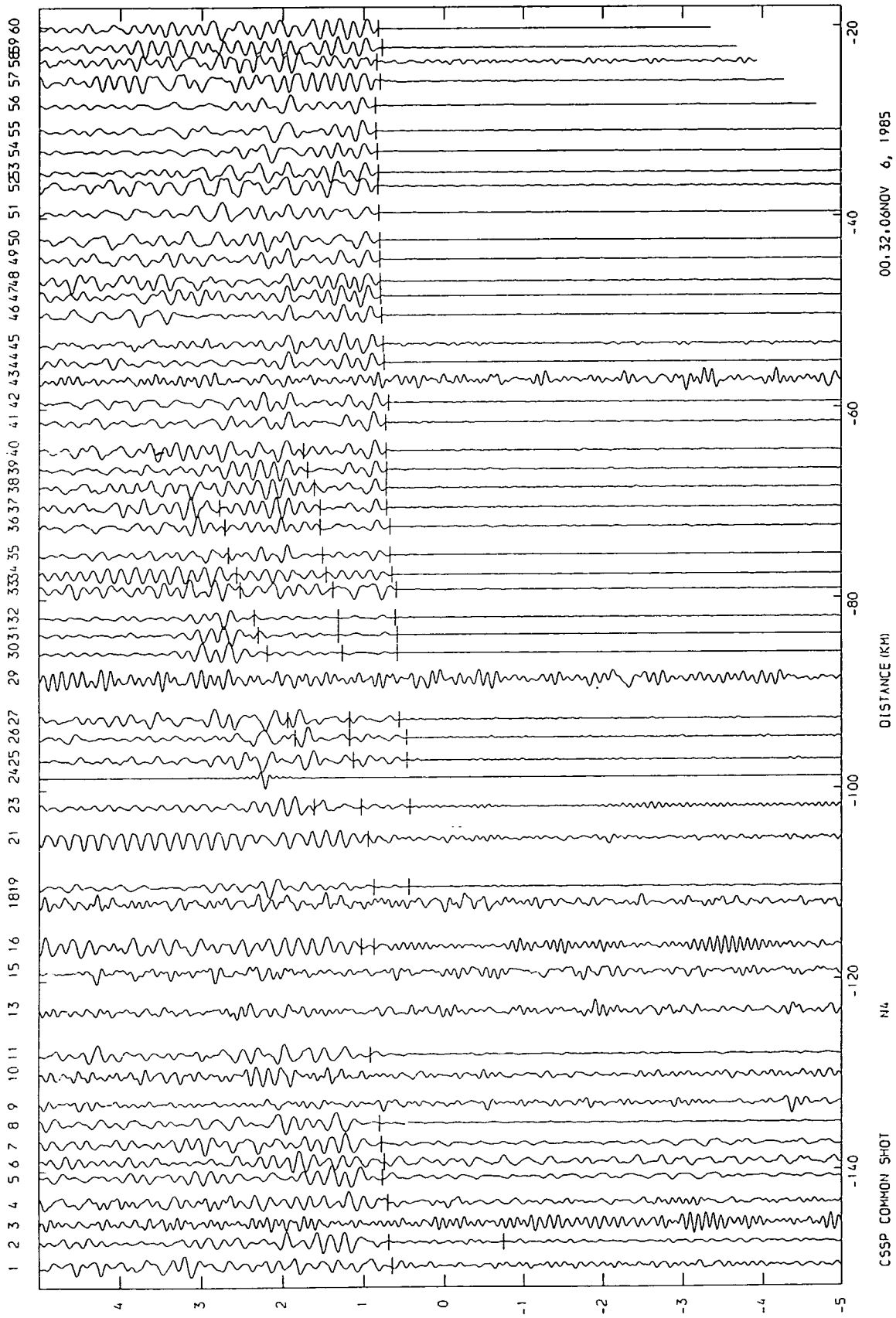
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



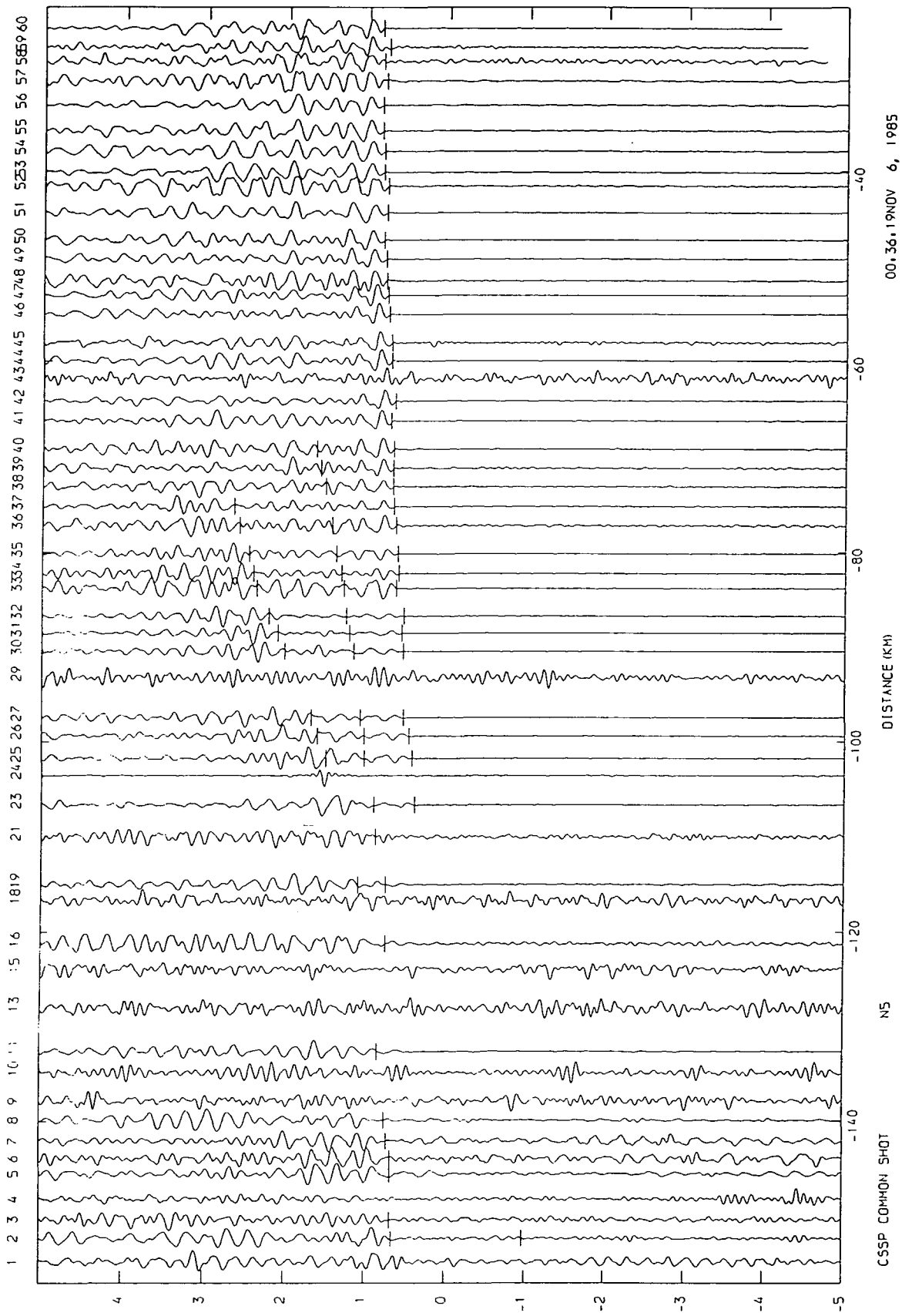
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



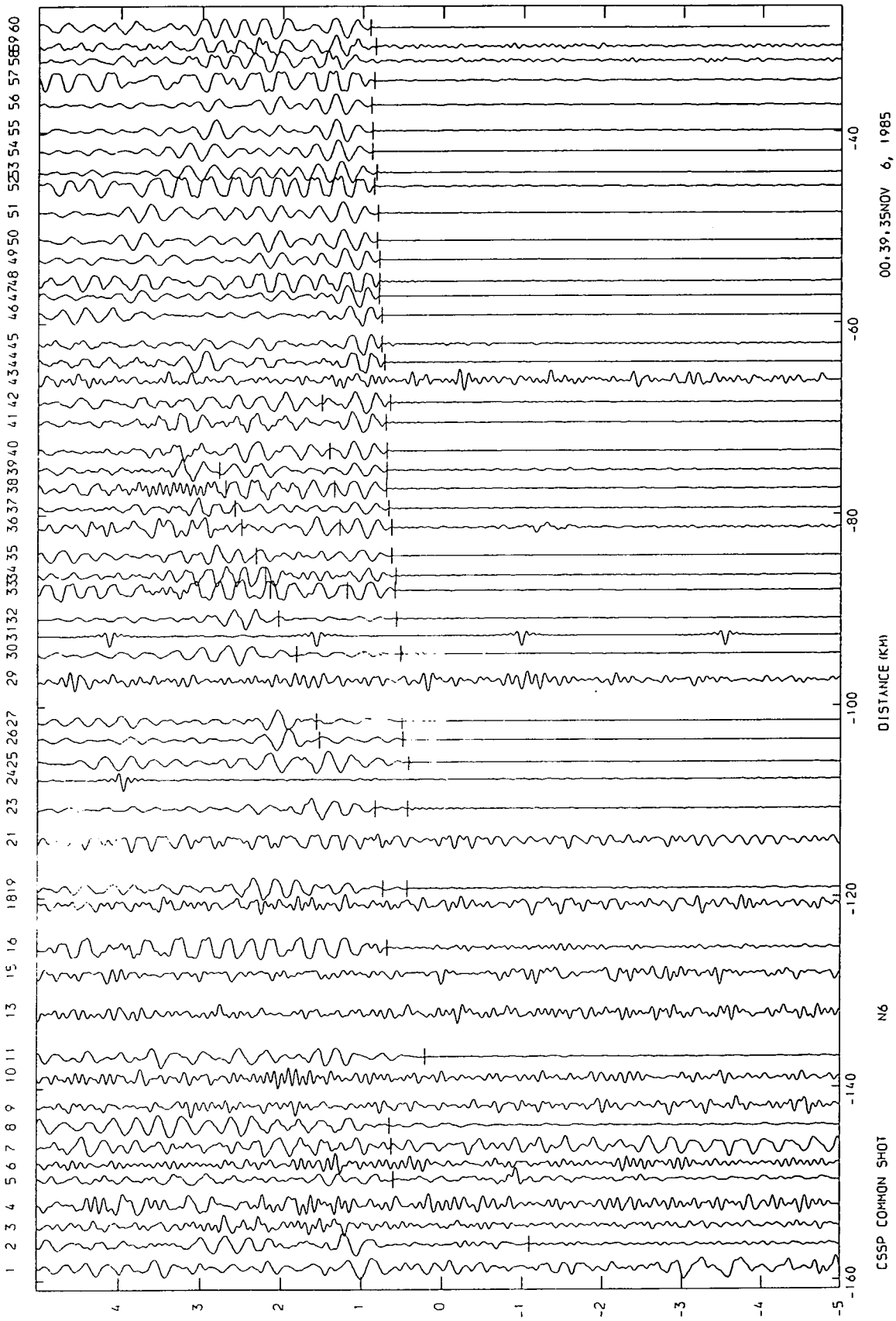
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



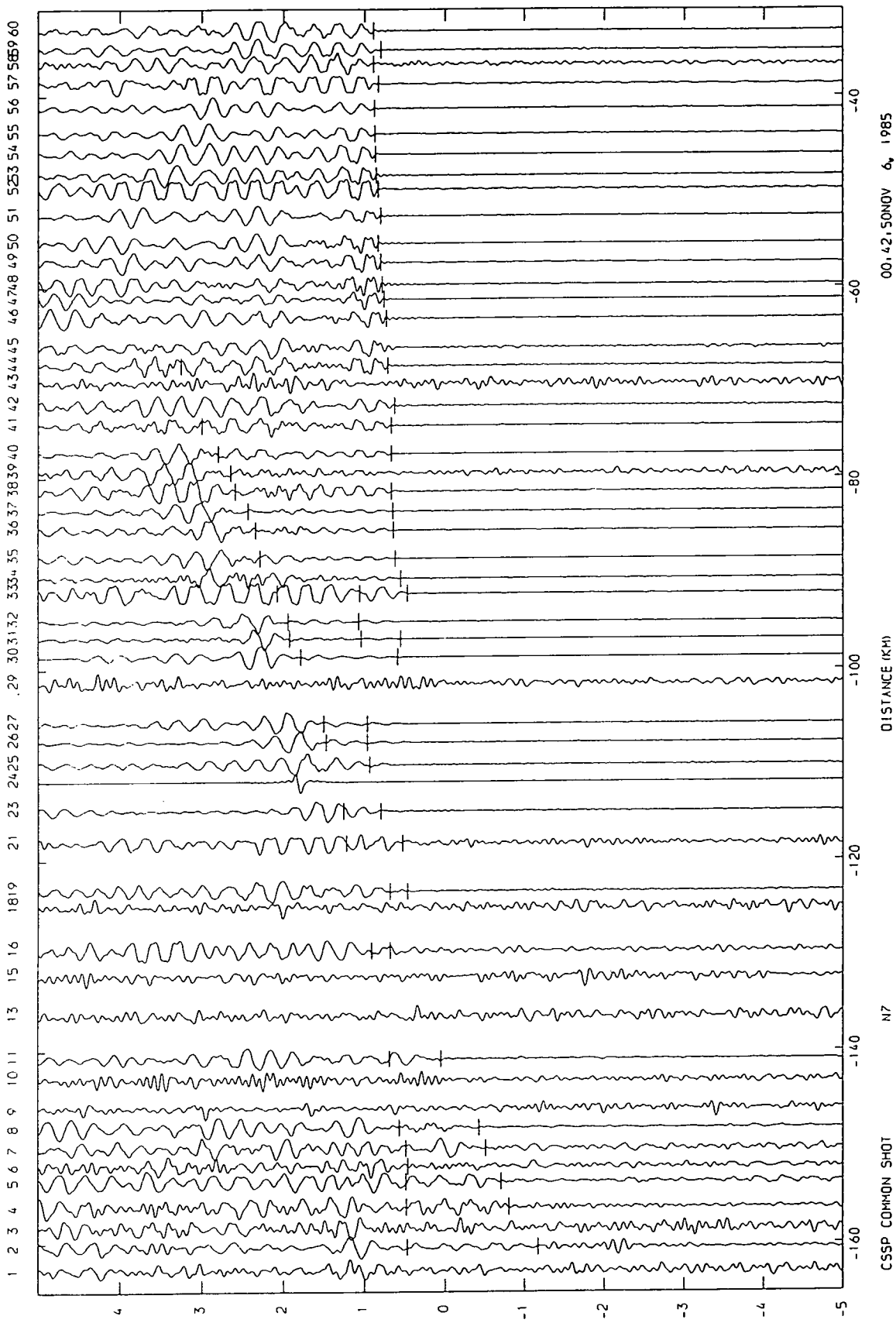
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



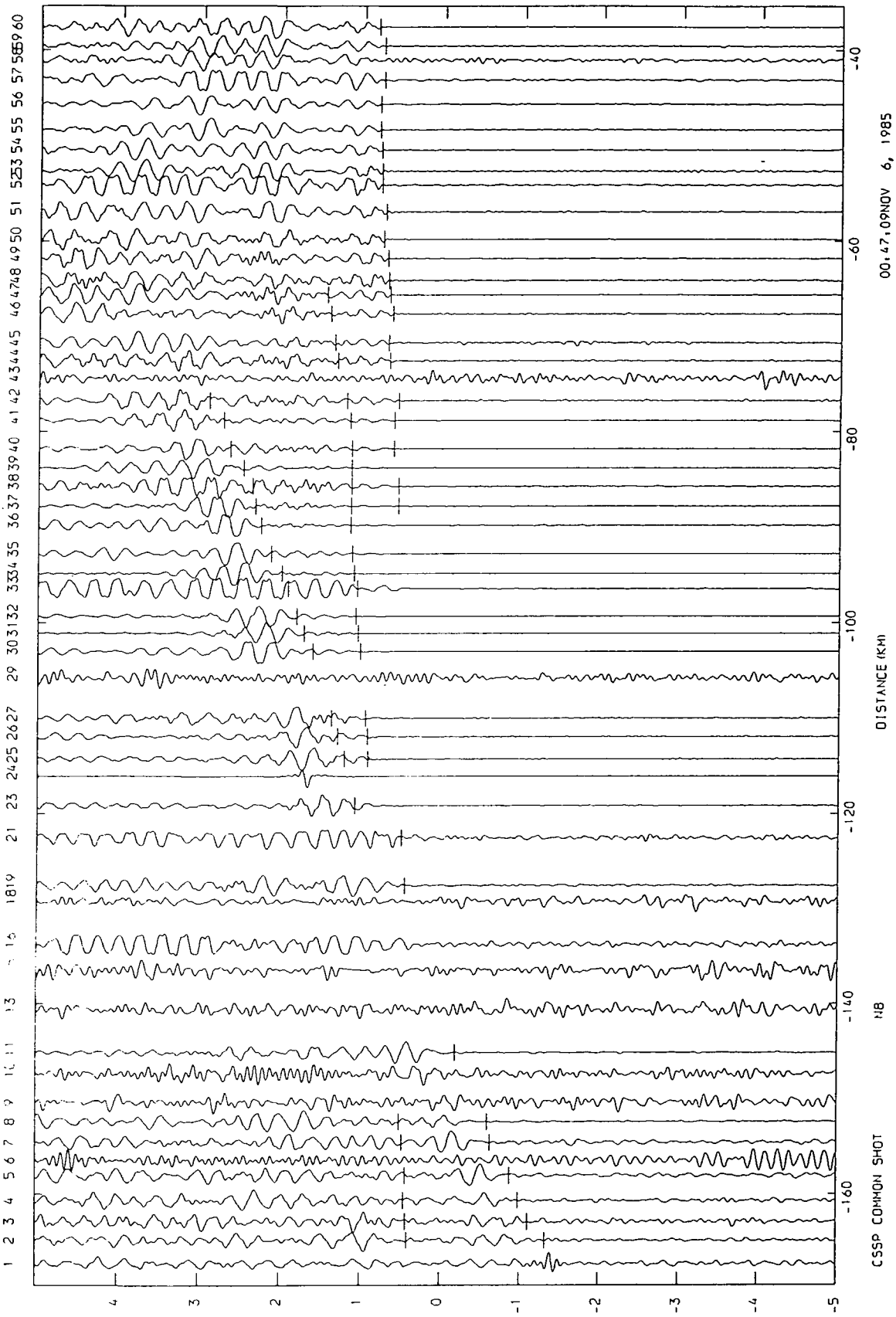
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



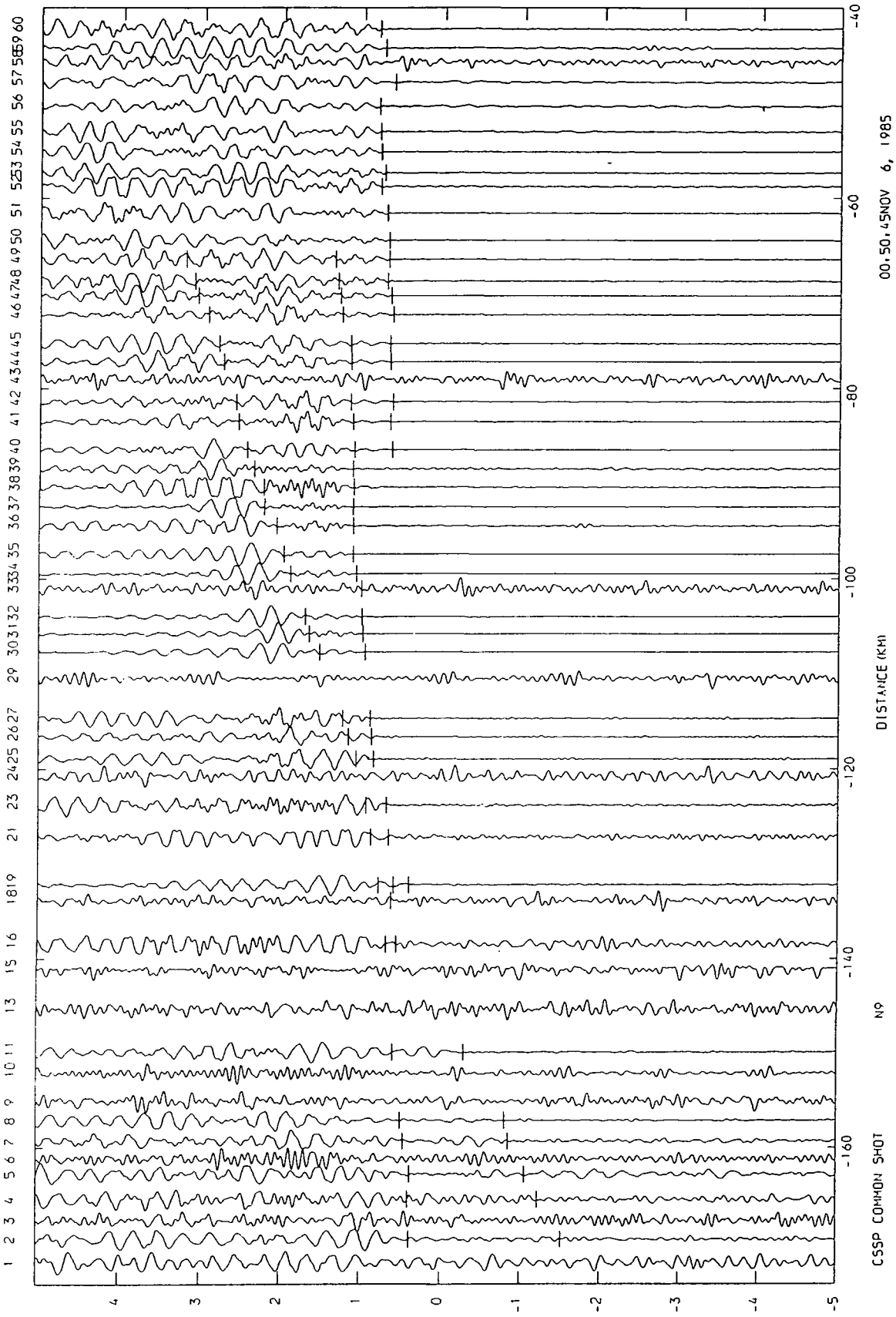
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



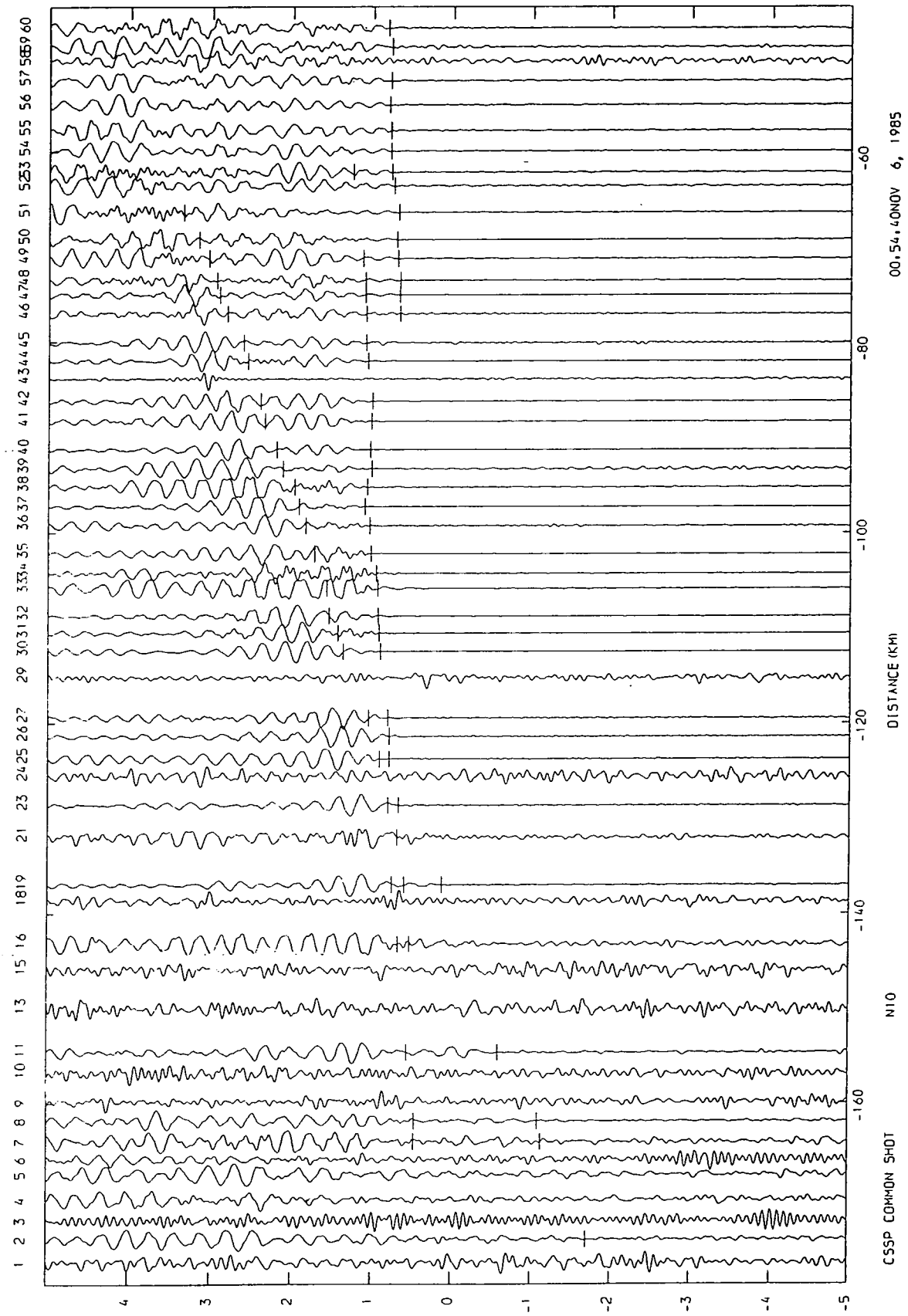
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



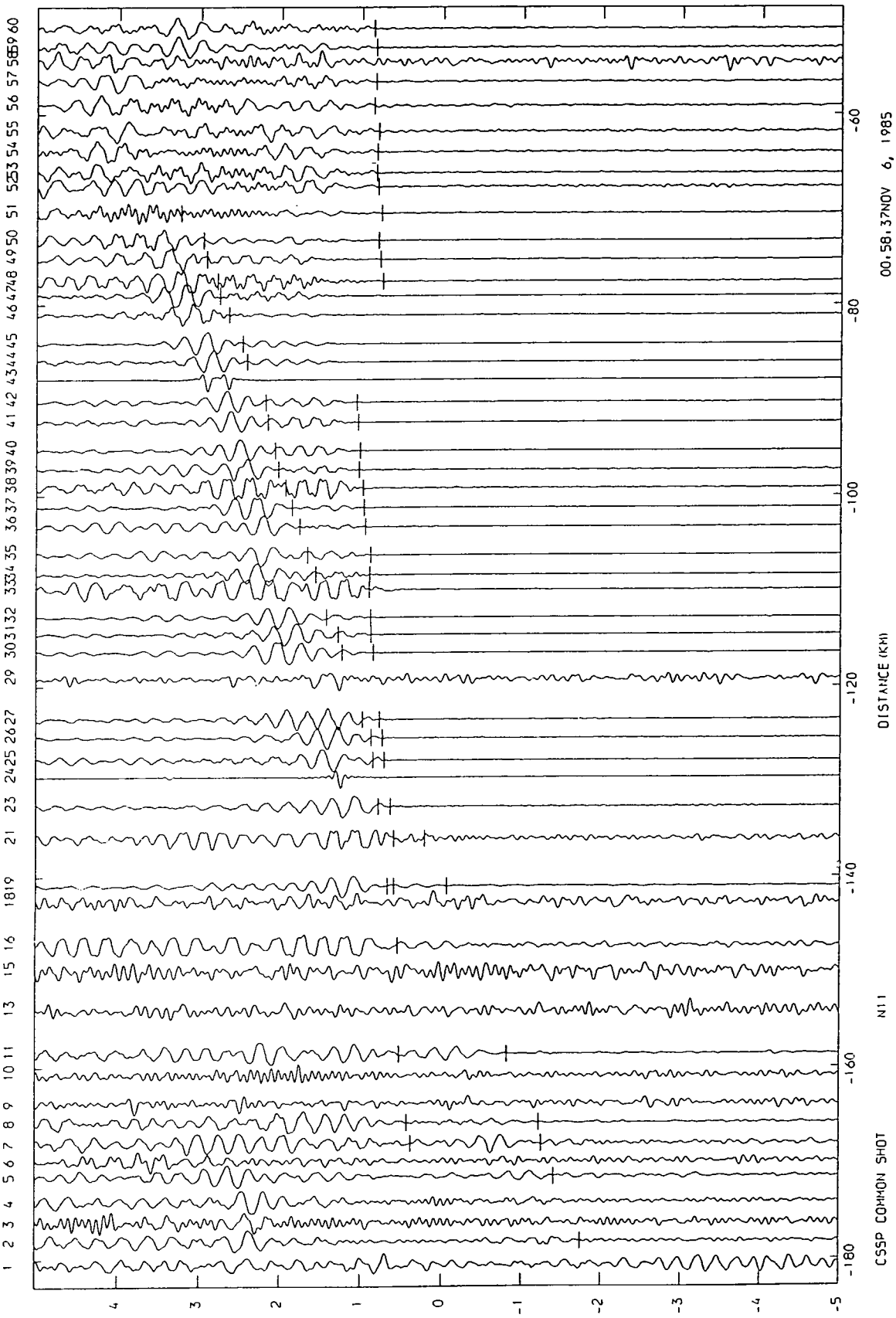
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



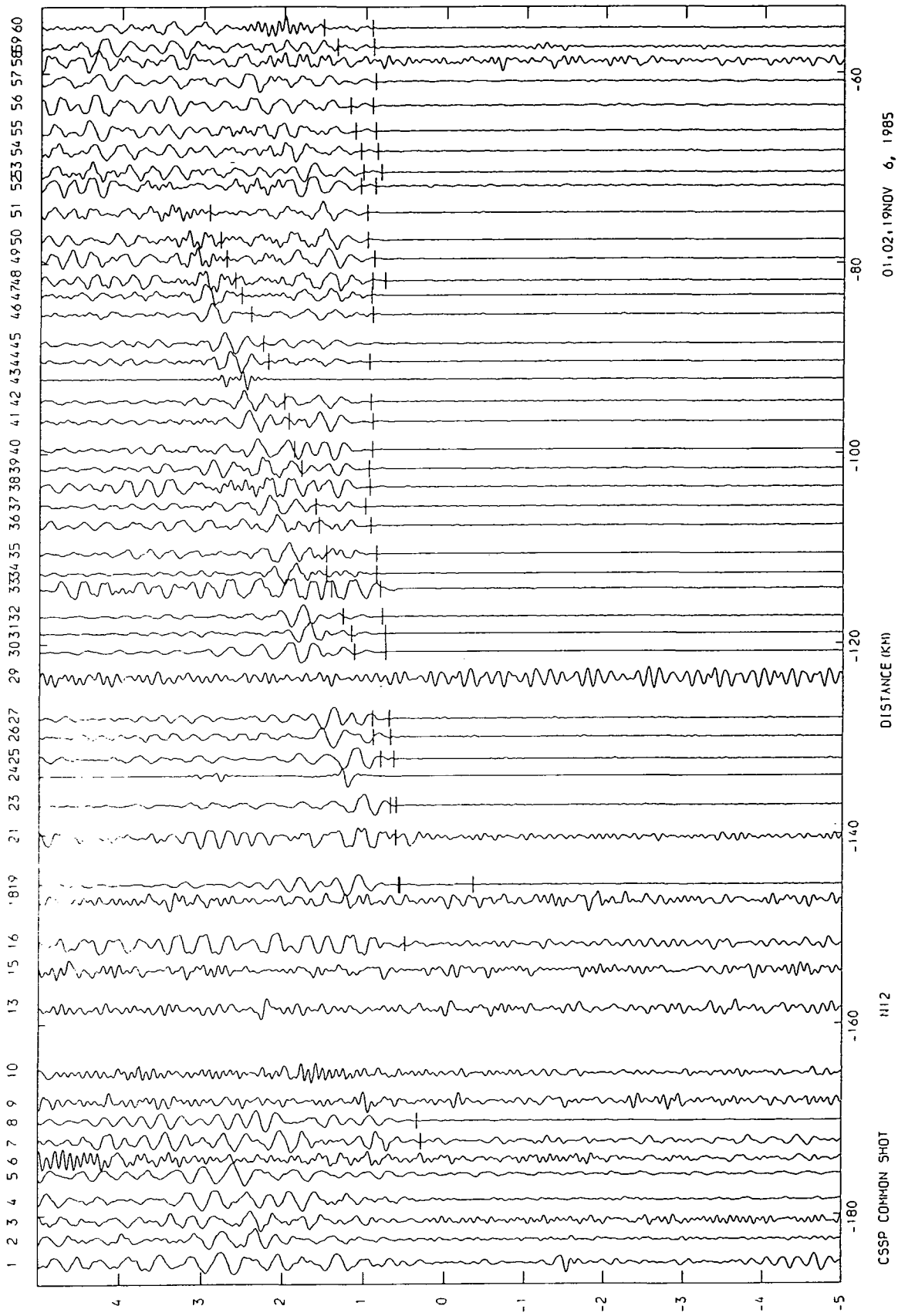
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



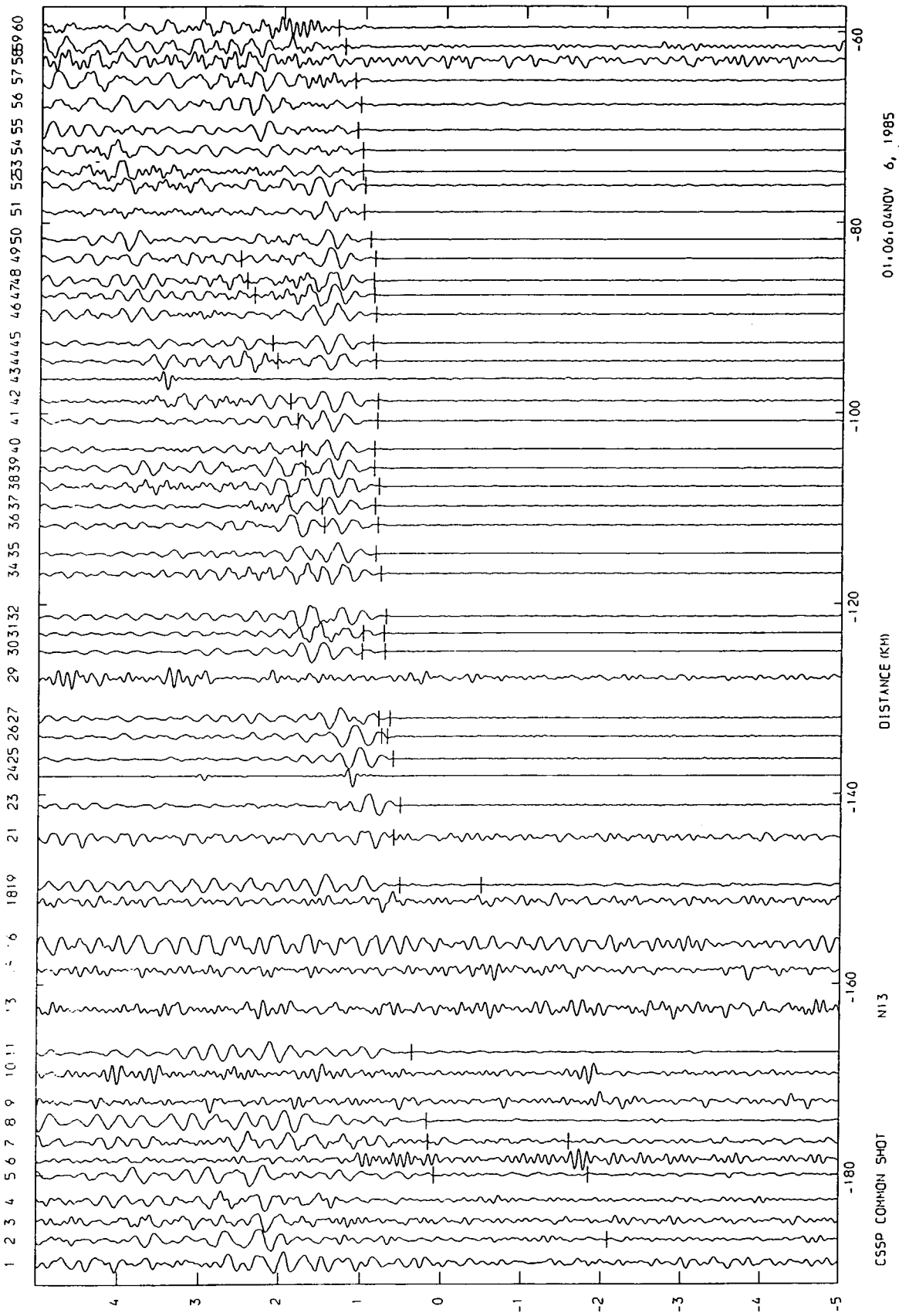
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



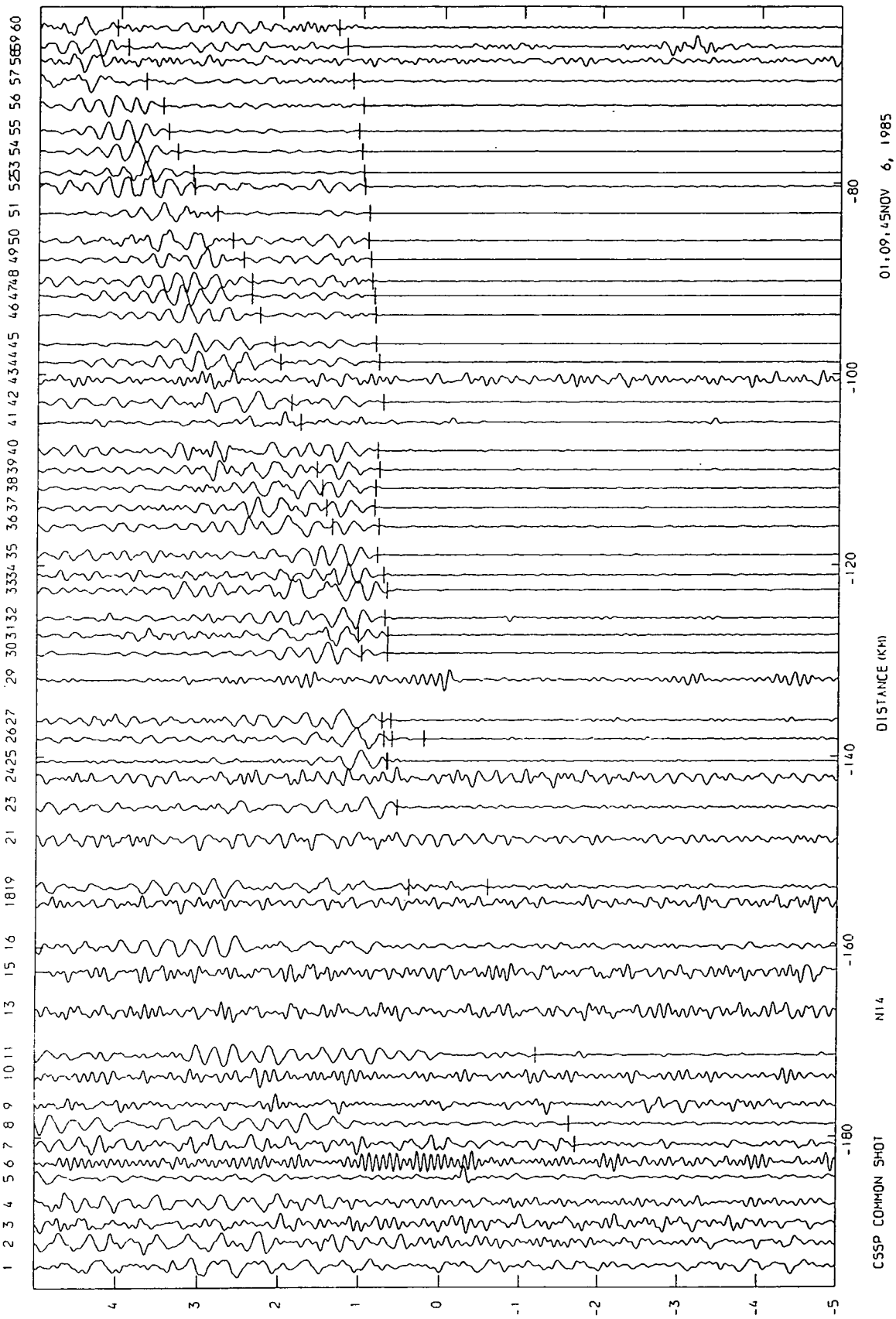
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

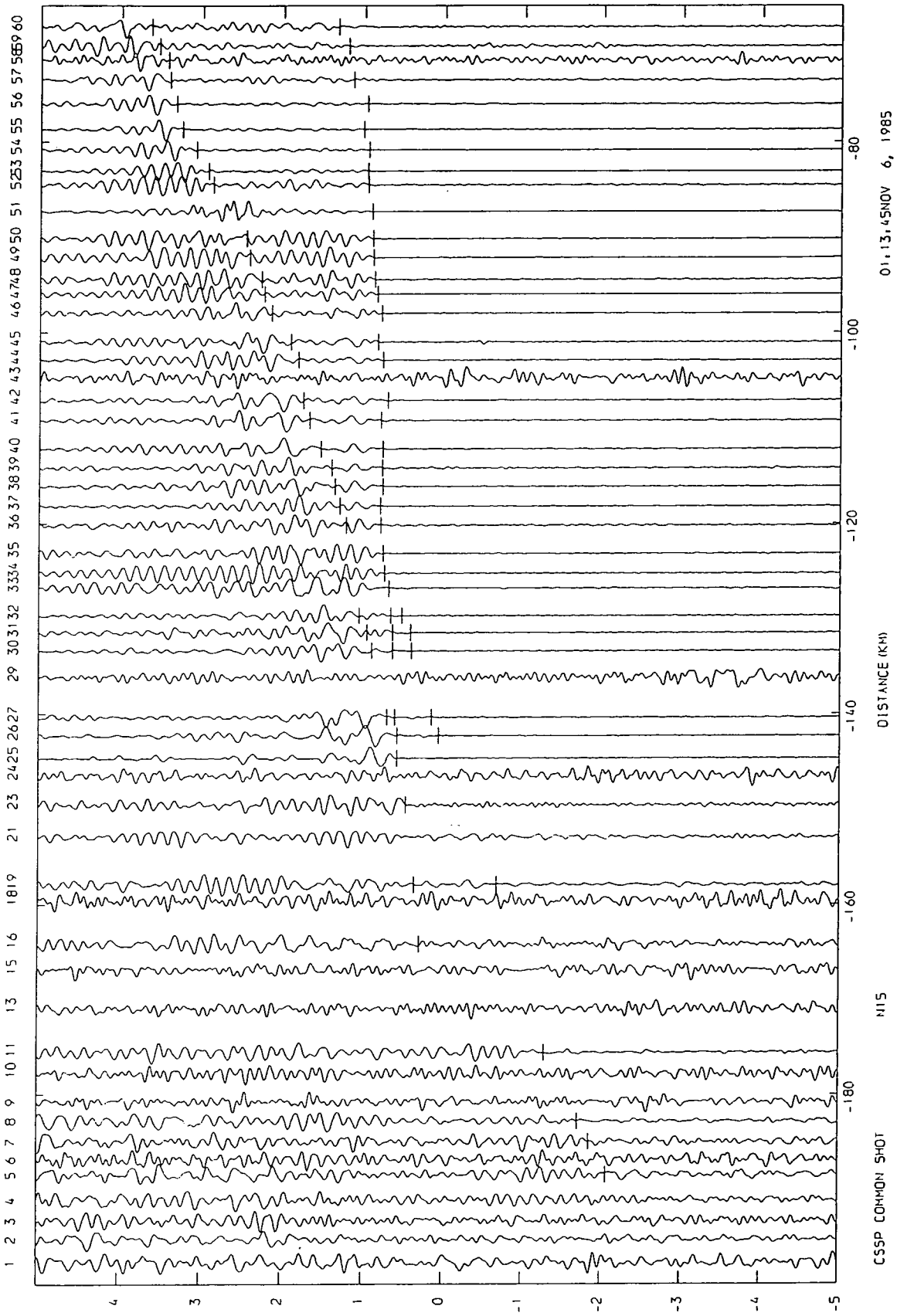


FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED

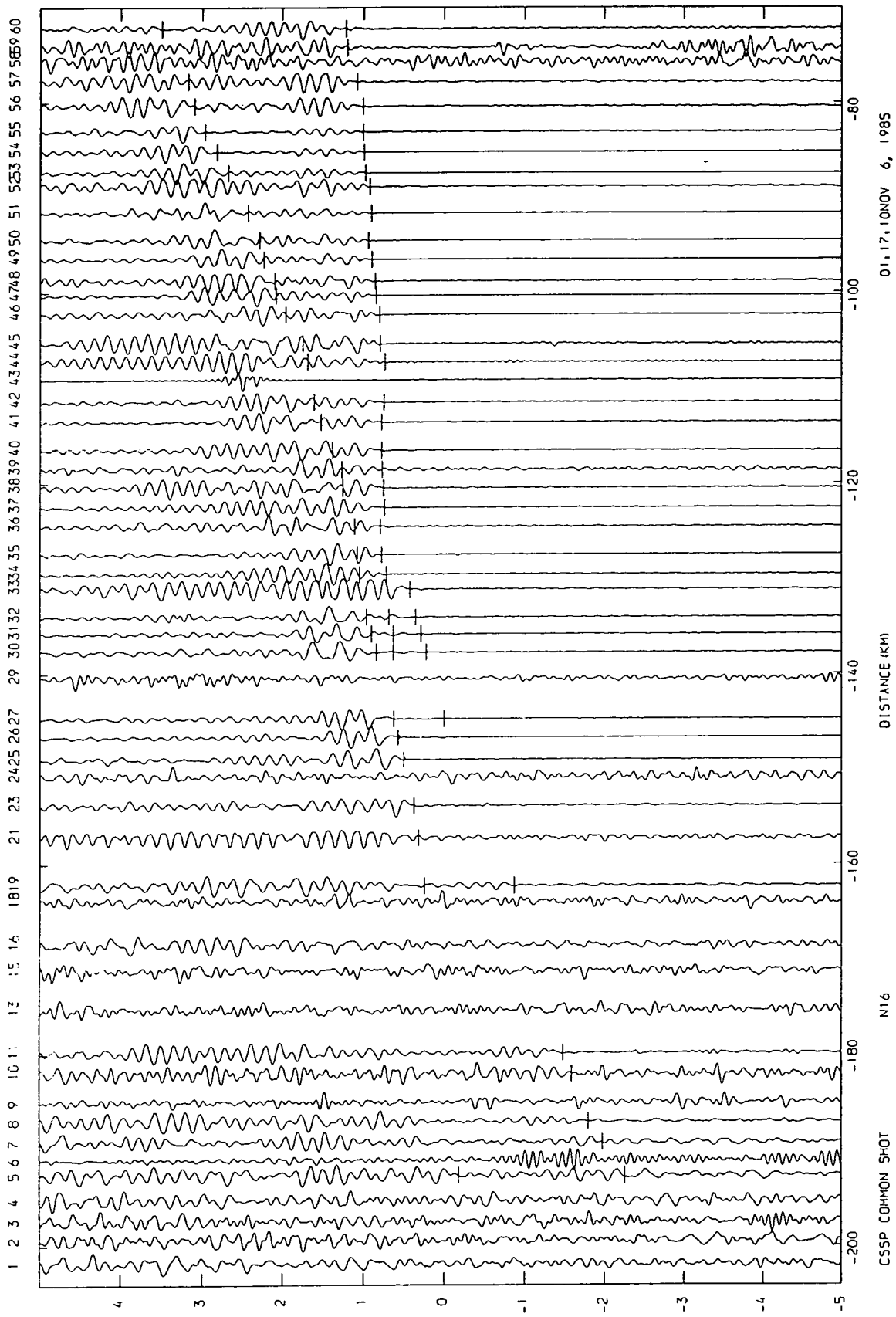


CSSP COMMON SHOT N14 01:09:45NOV 6, 1985

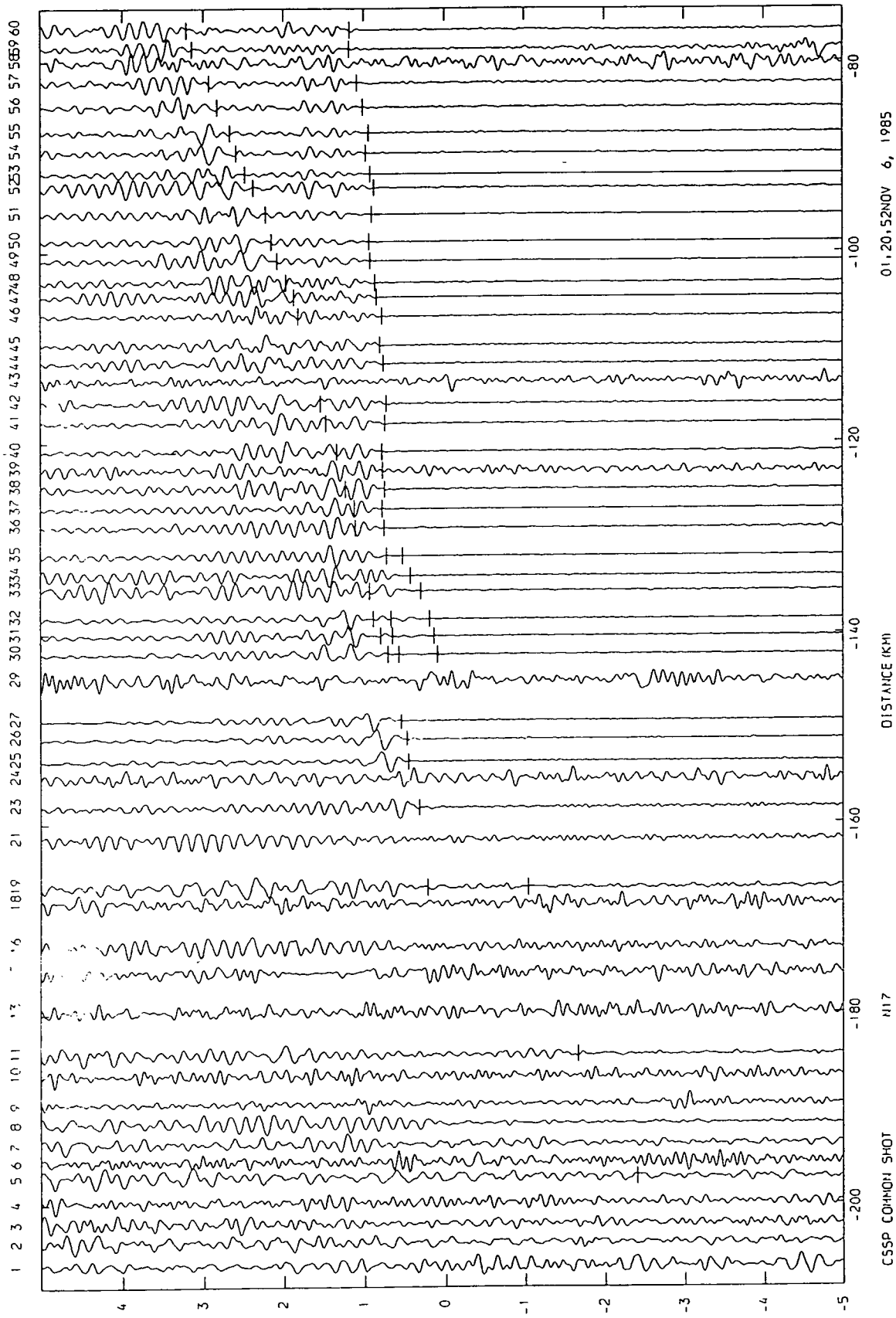
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



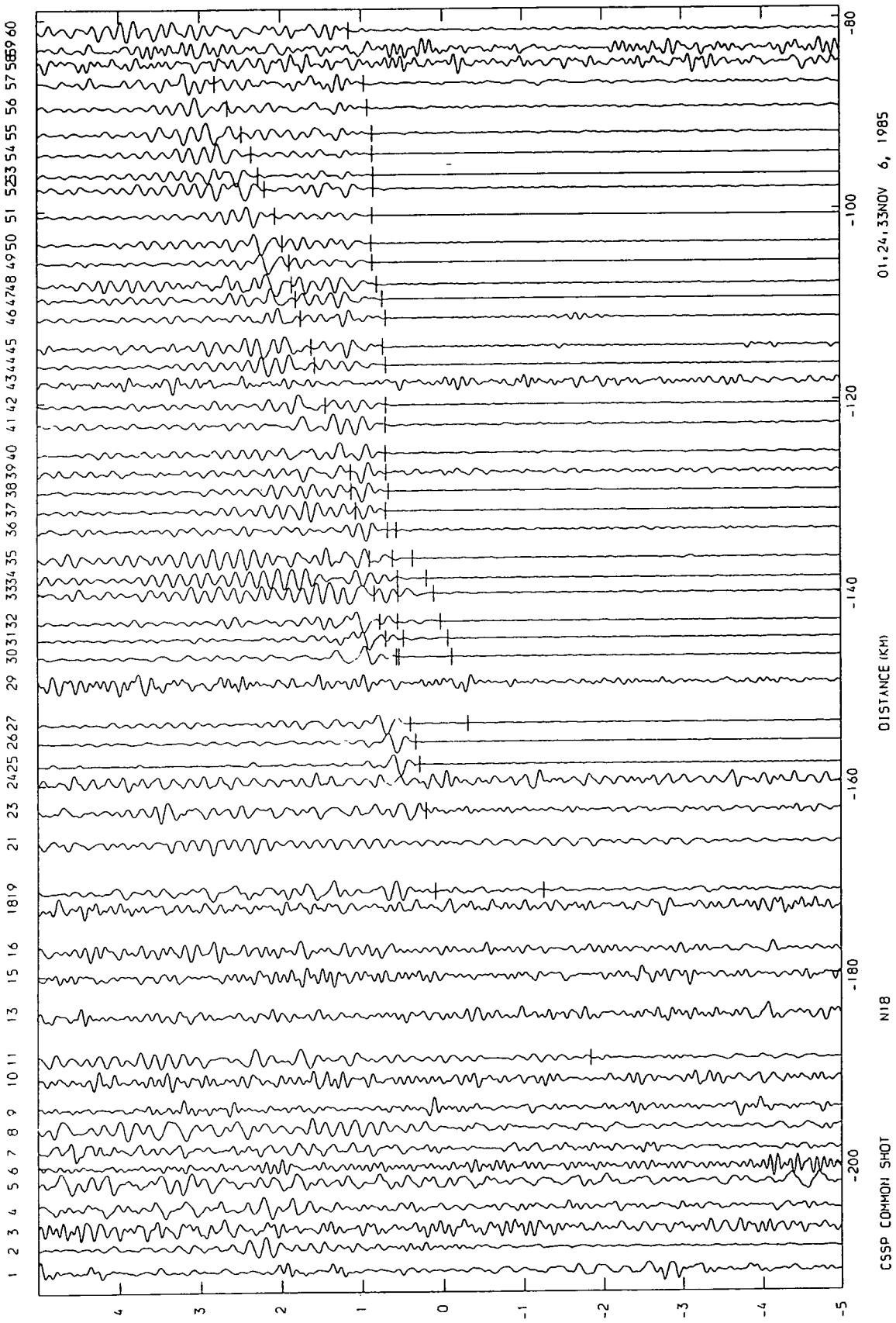
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED

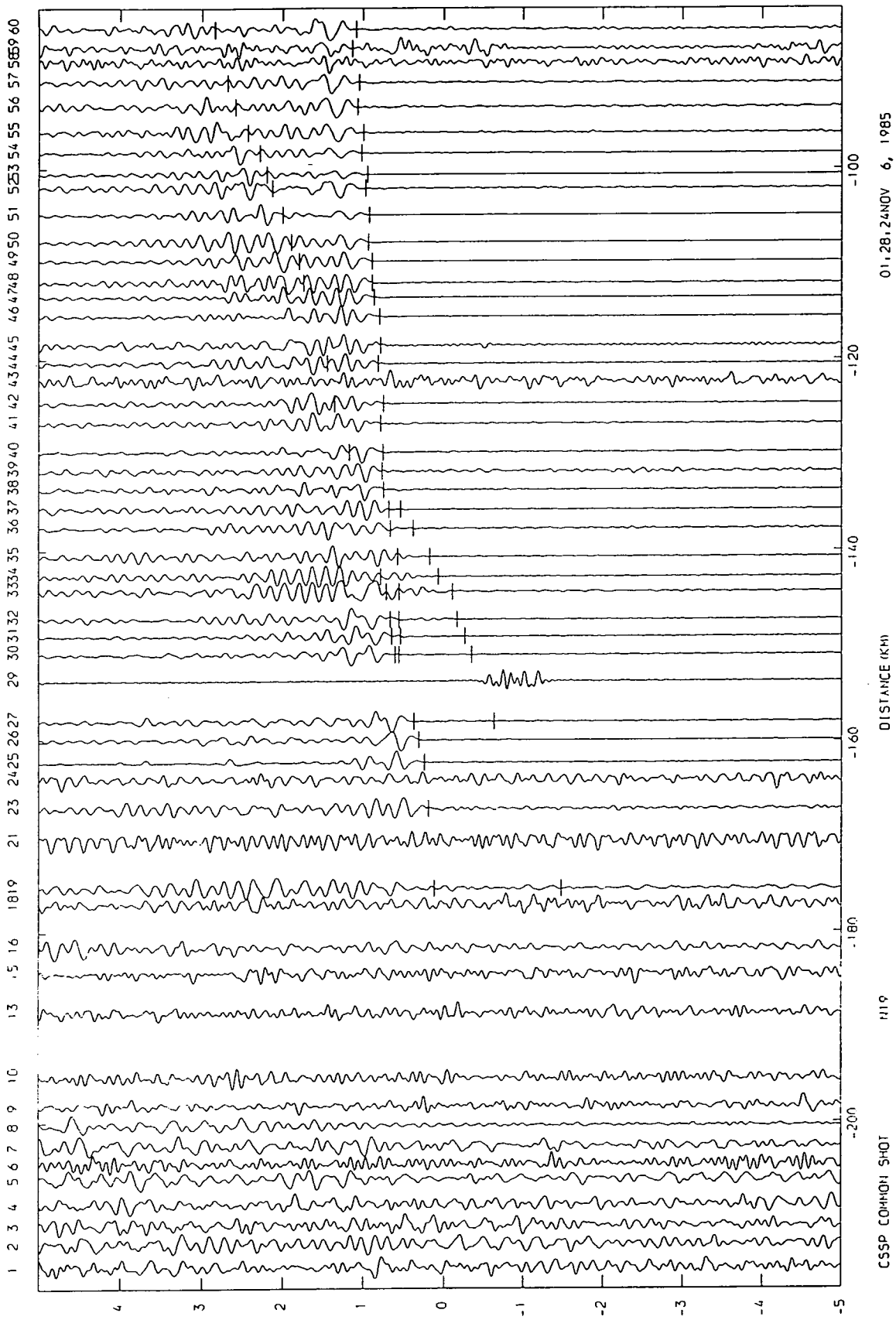


FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

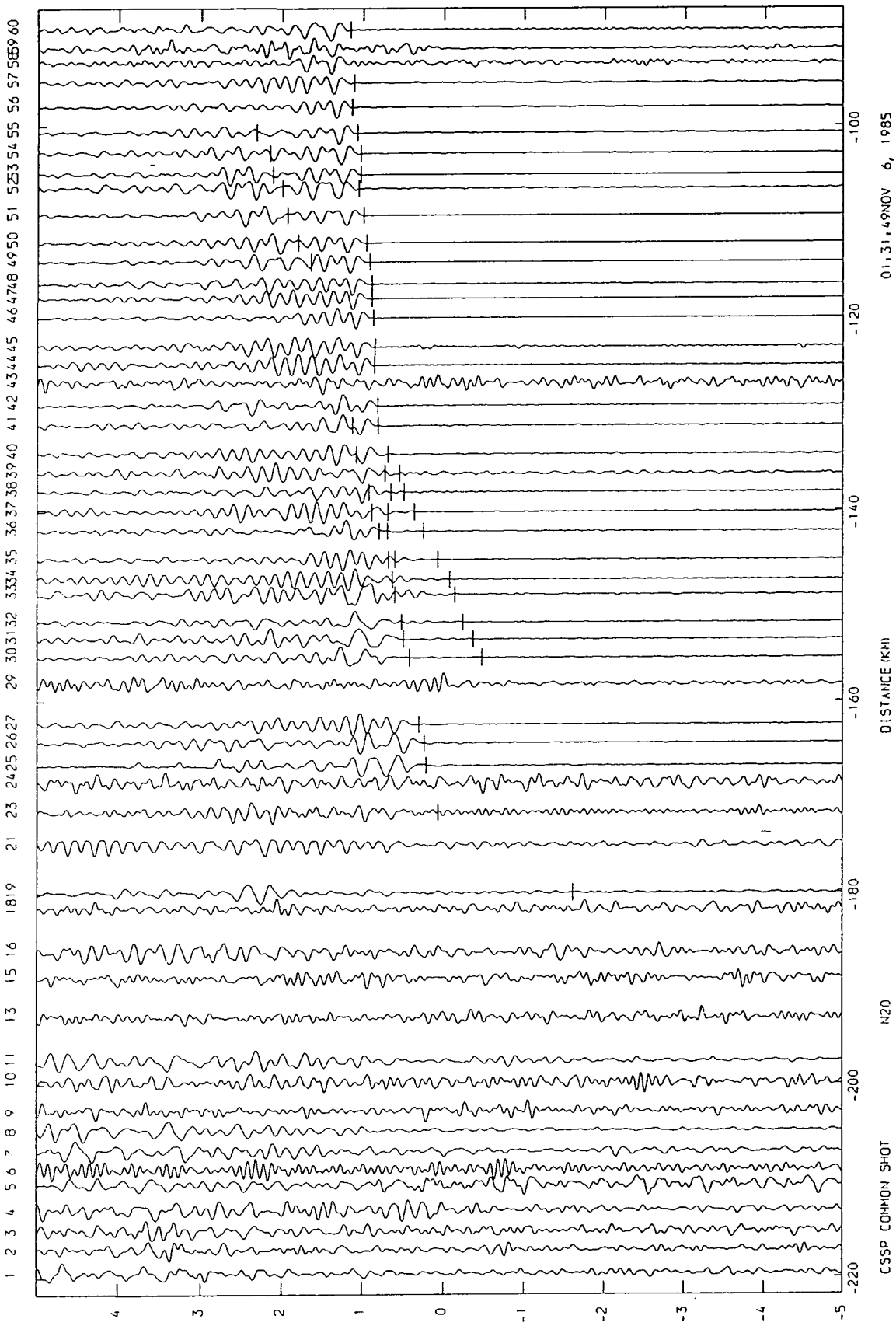


CSSP COMMON SHOT N18 01:24:33NOV 6, 1985

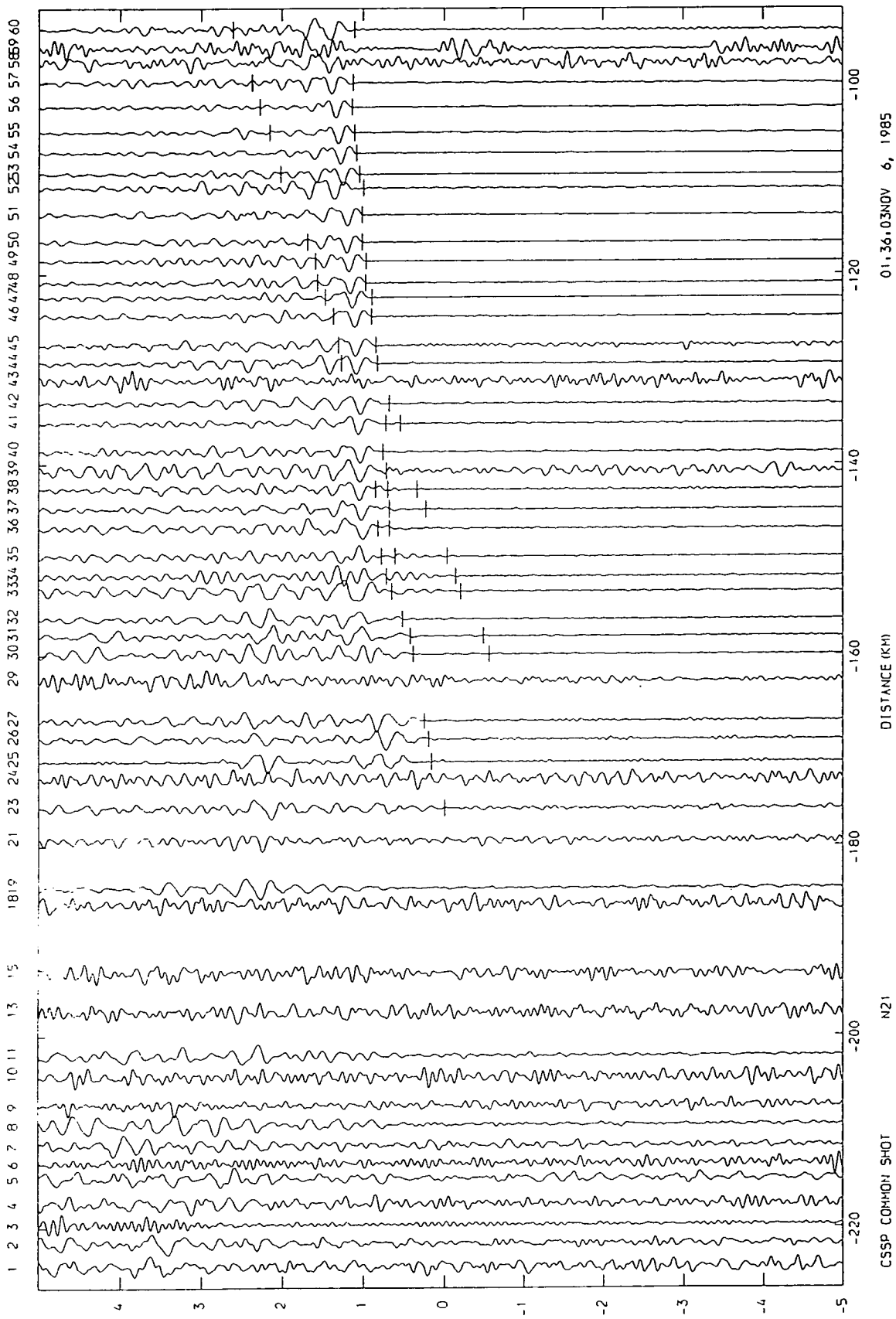
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



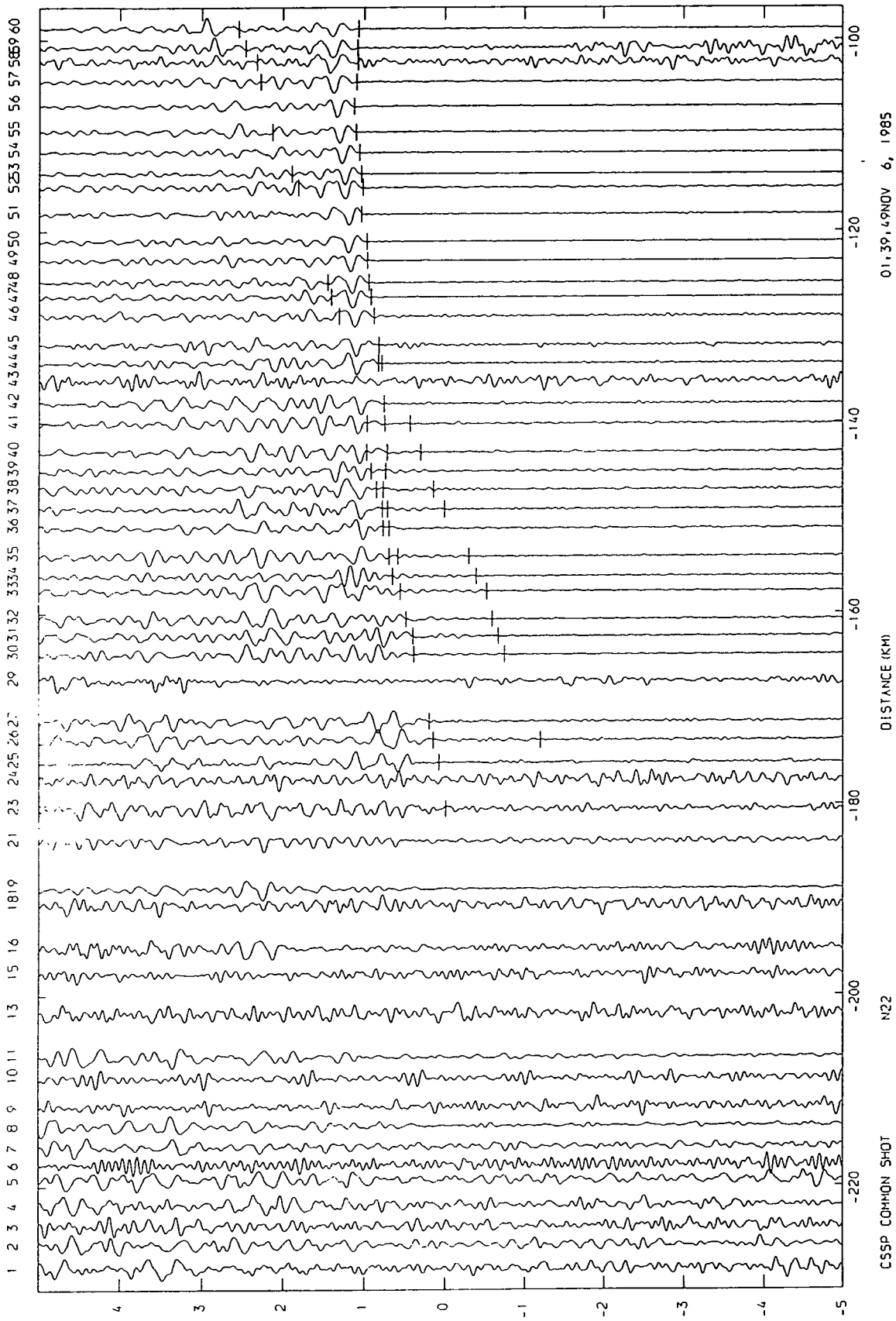
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



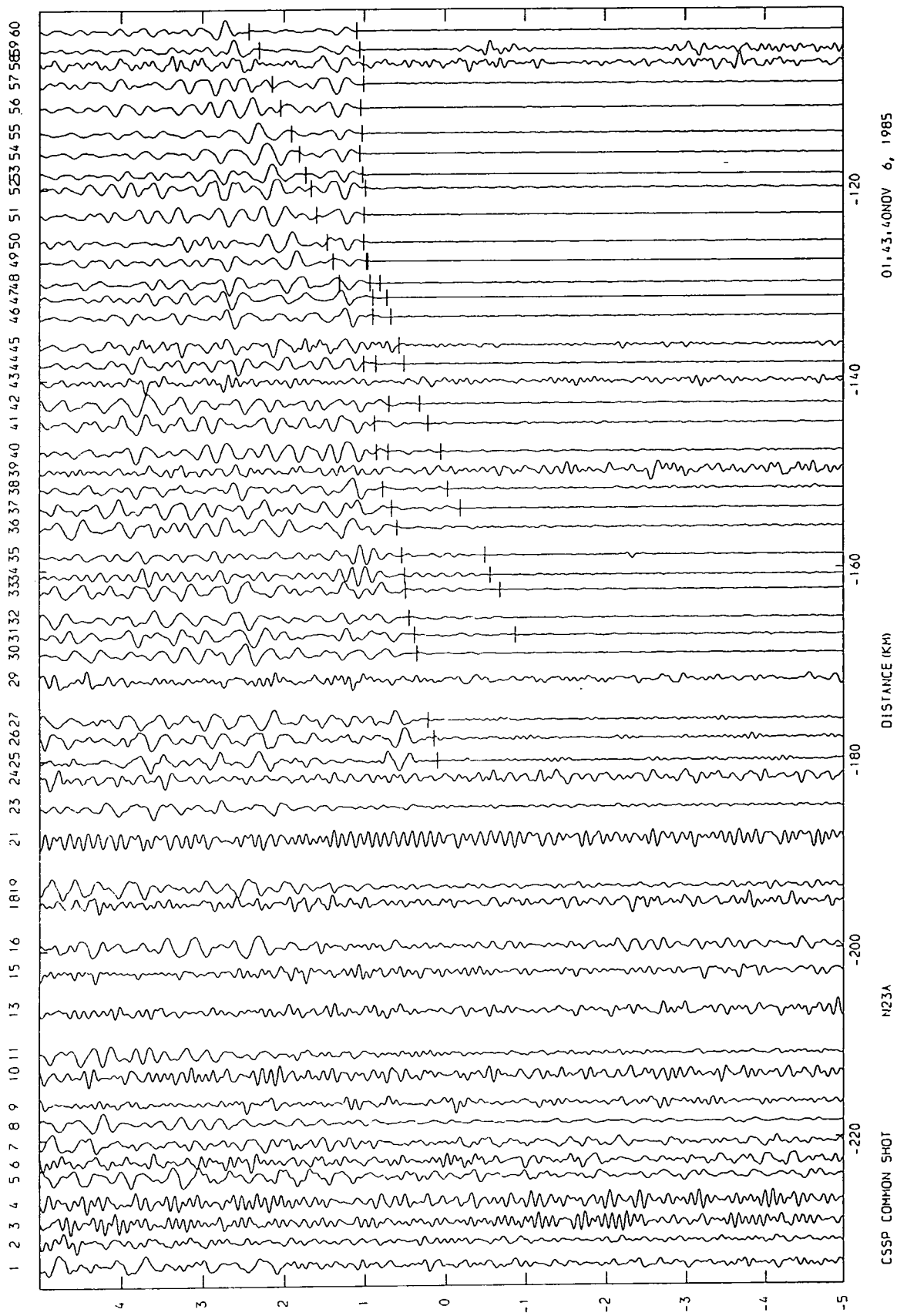
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



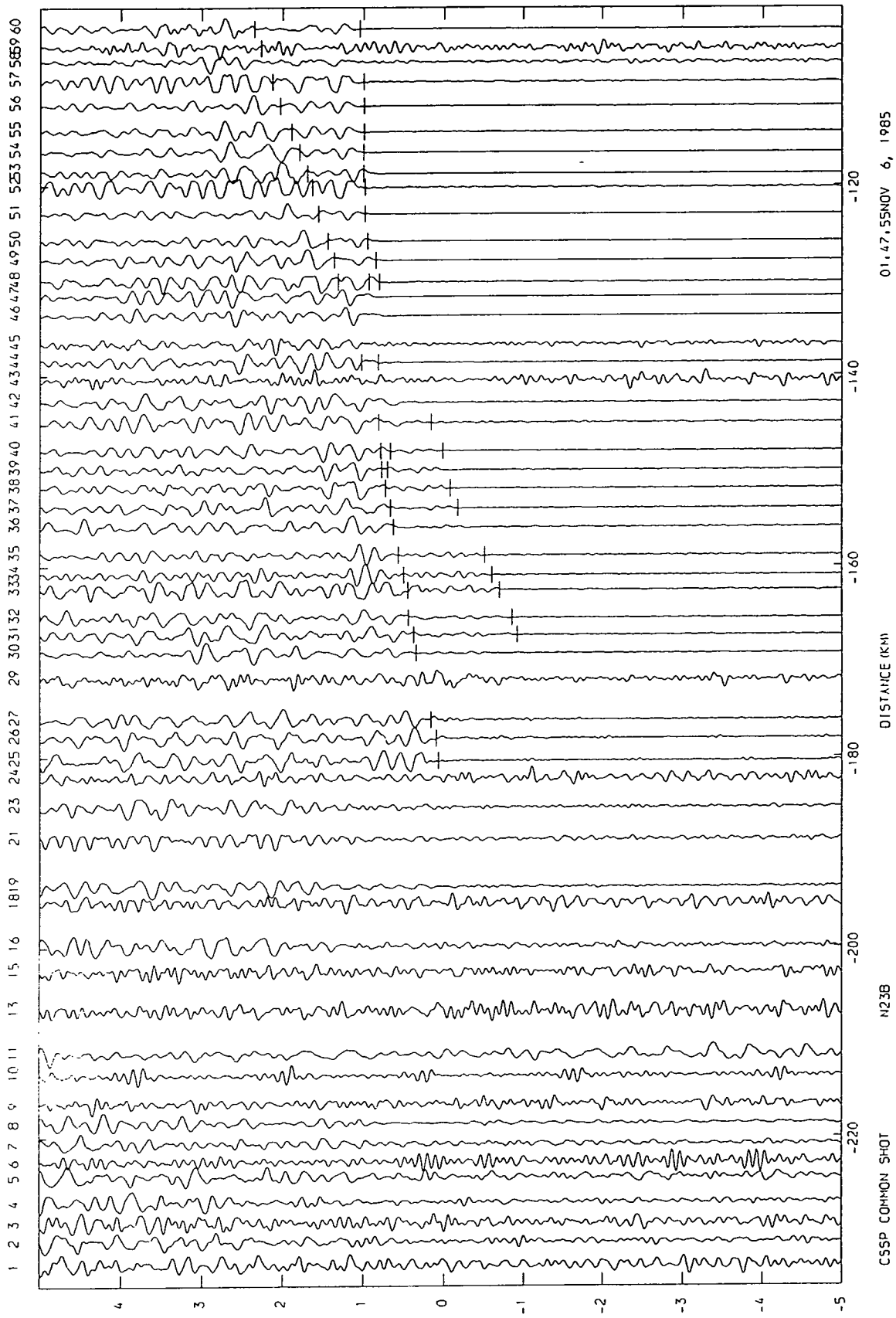
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



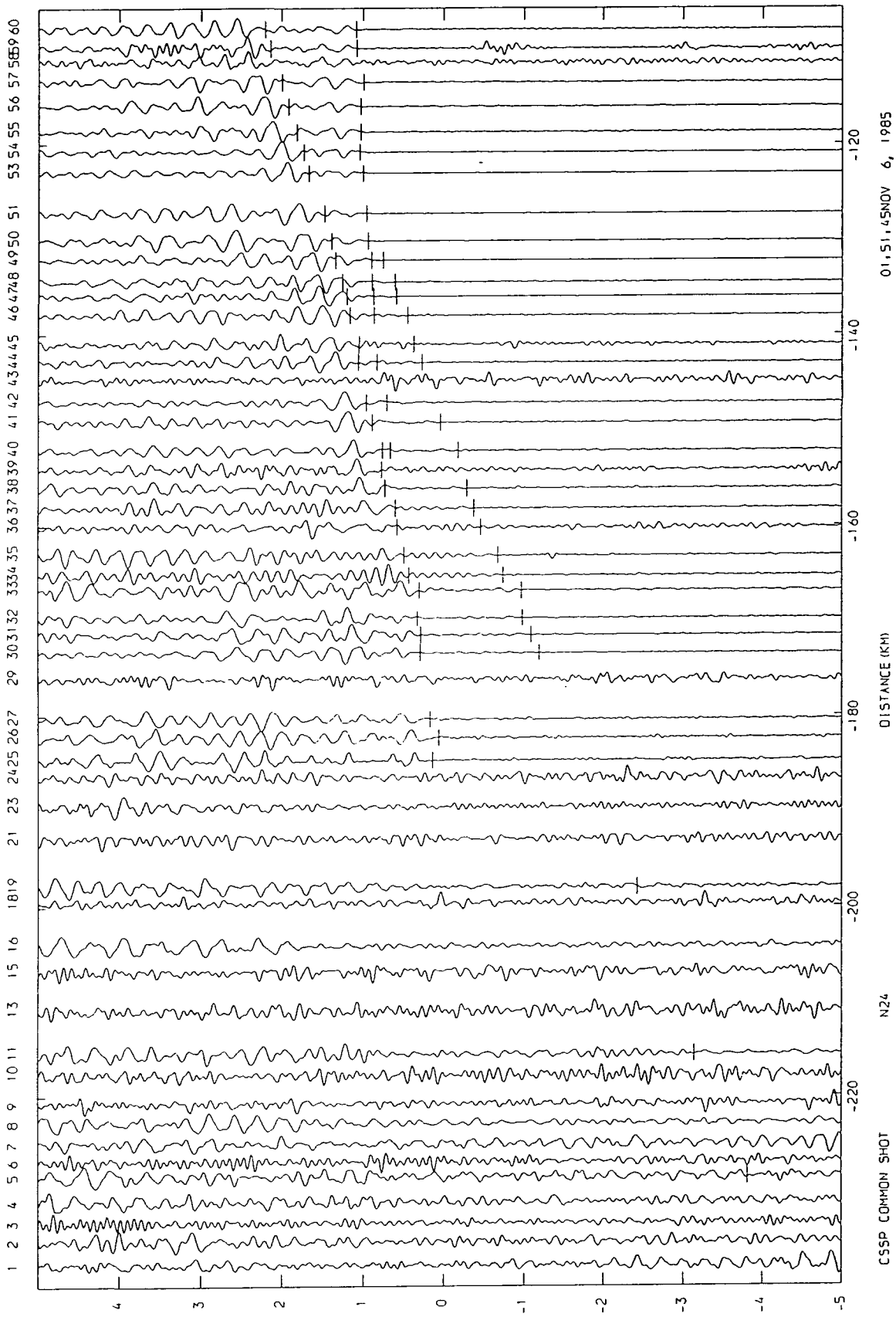
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



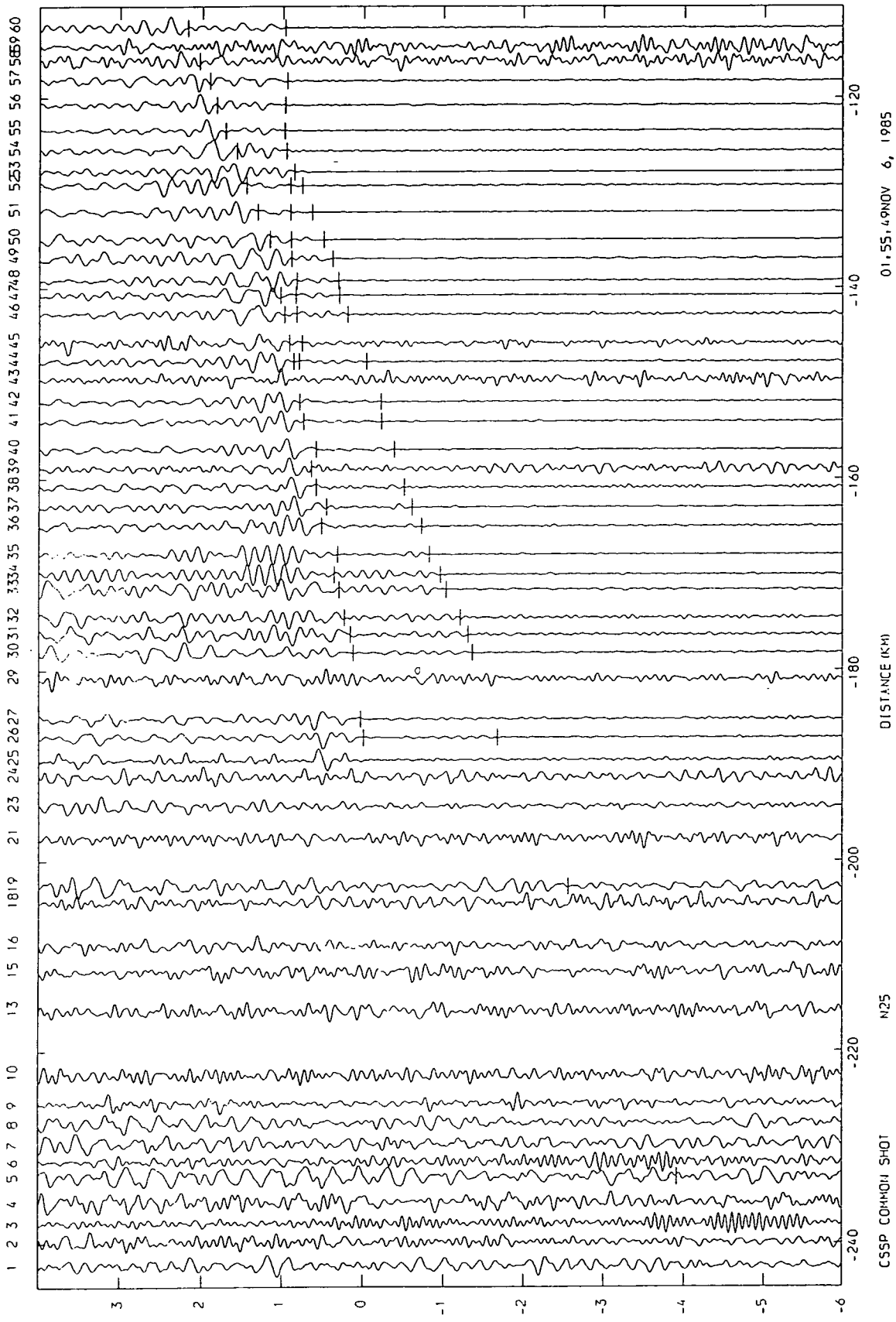
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



CSSP COMMON SHOT N24

01,51,45NOV 6, 1985

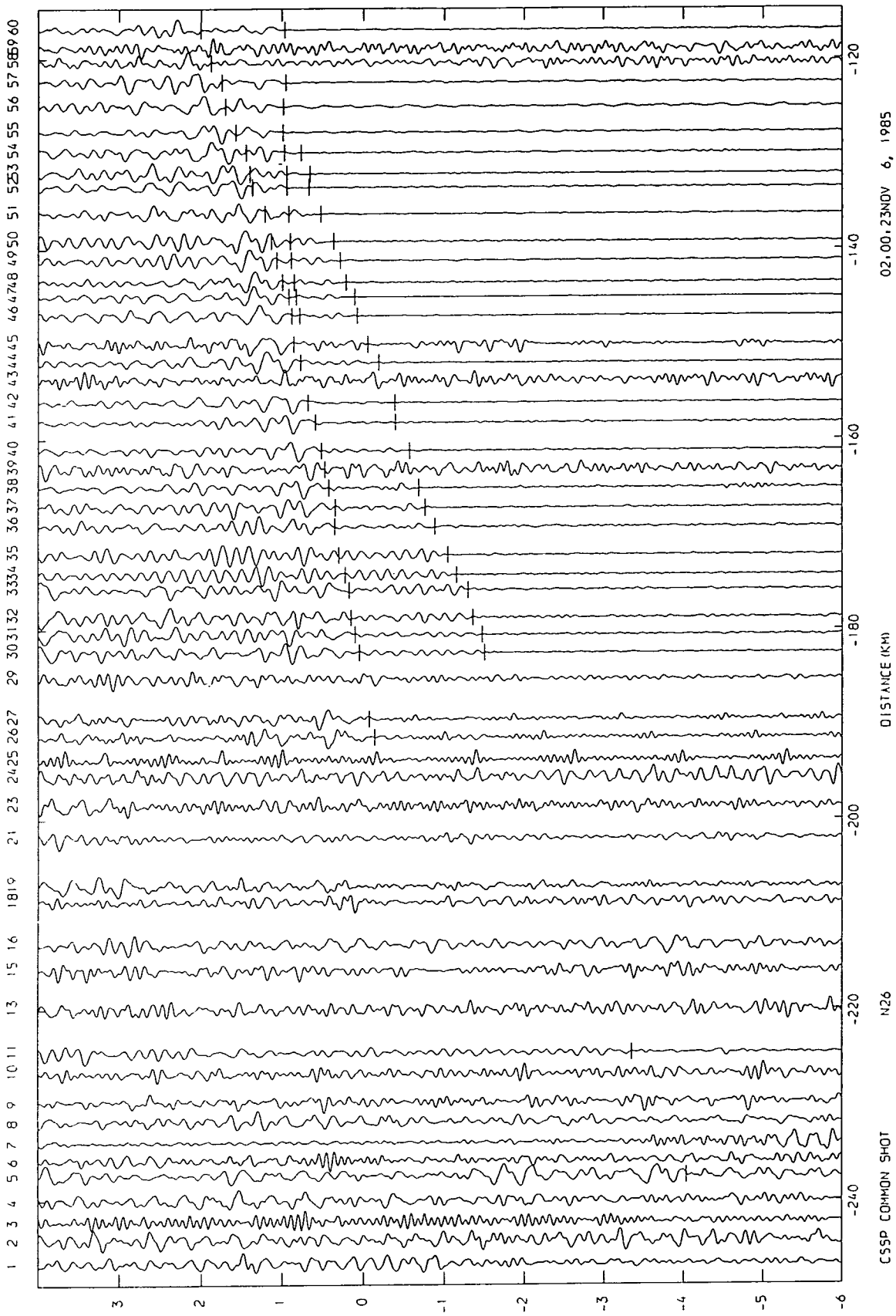
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



CSSP COMMON SHOT N25

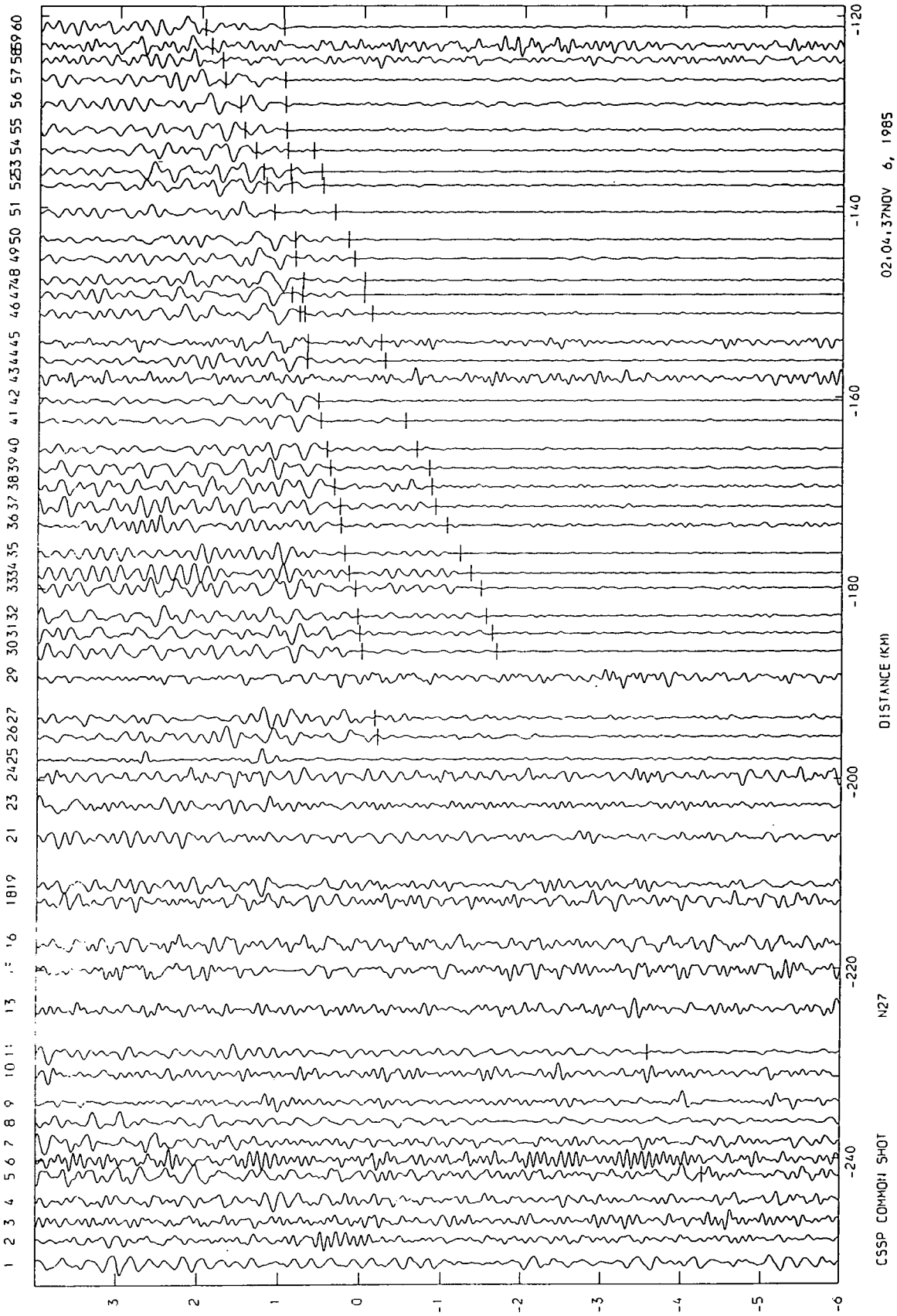
01:55:49NOV 6, 1985

FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

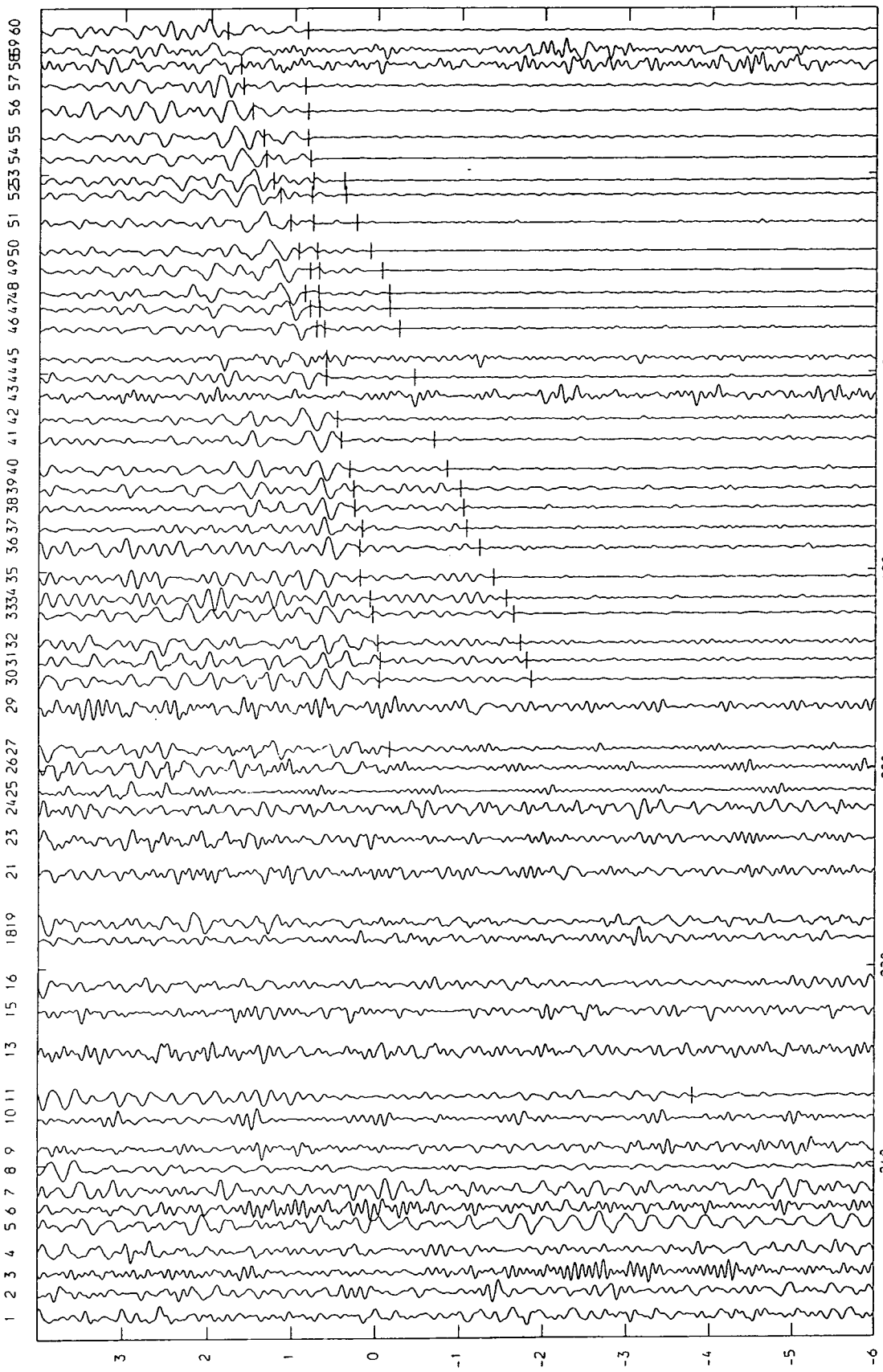


CSSP COMMON SHOT N26 02.00.23NOV 6, 1985

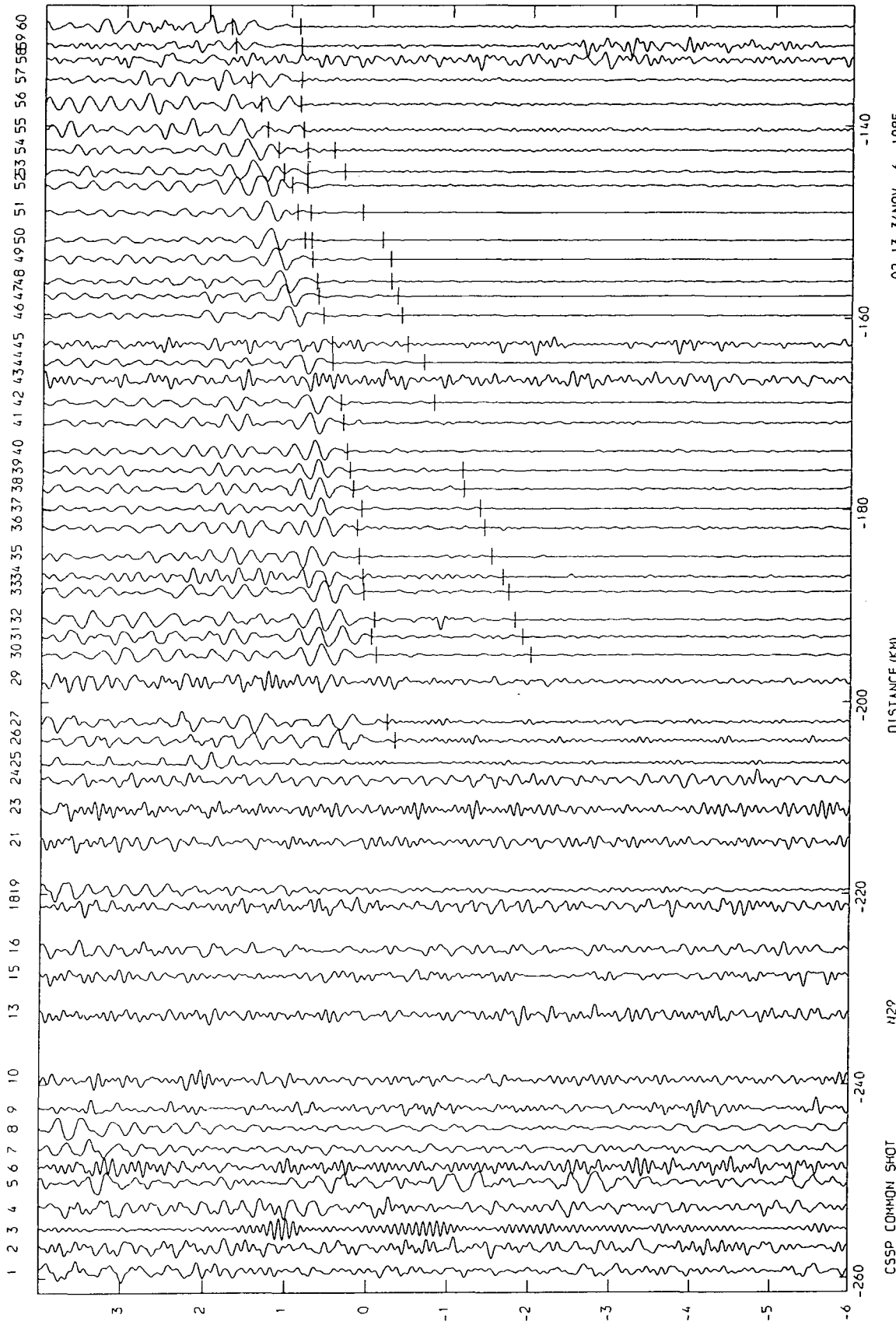
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



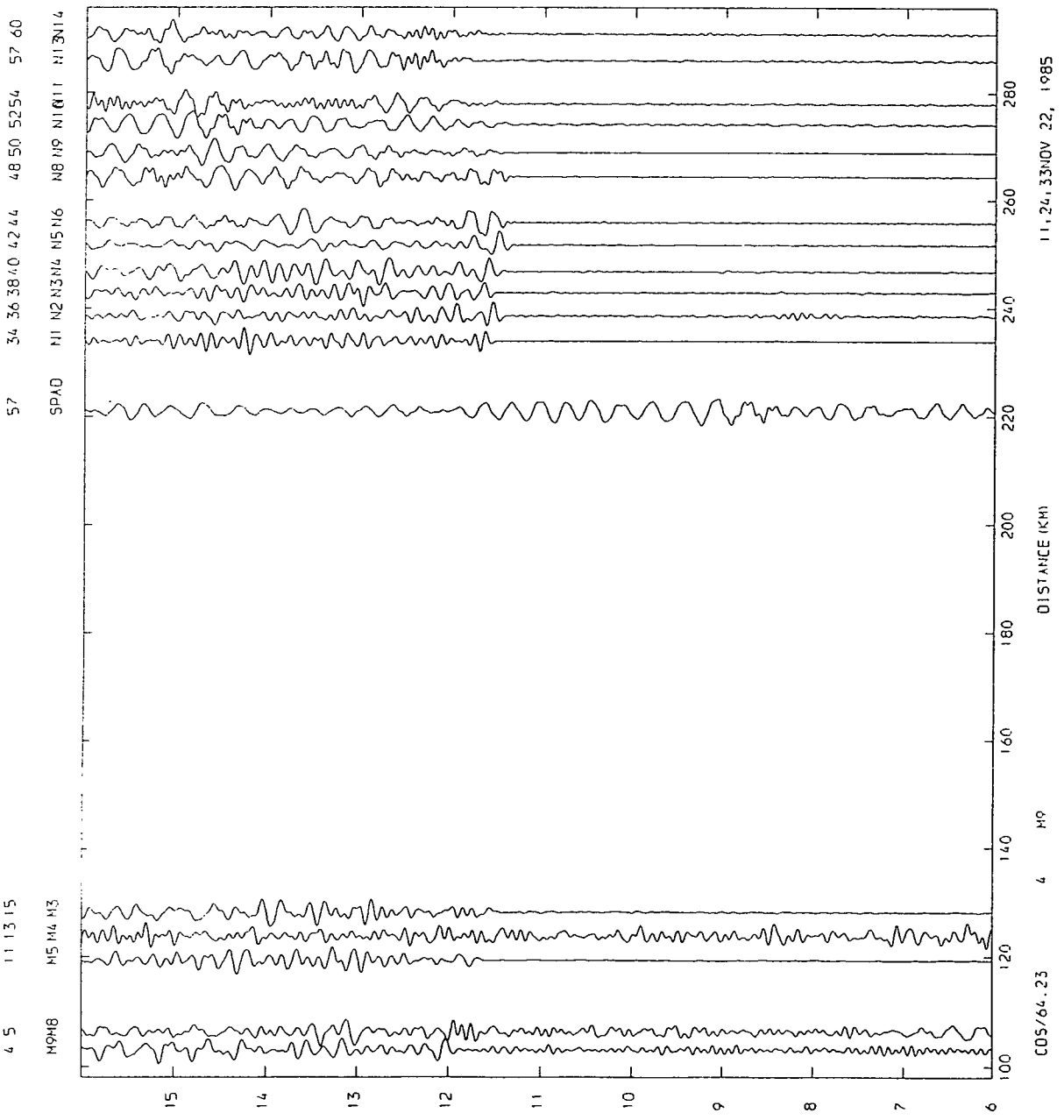
Appendix C

COMMON DISTANCE SECTIONS AND MISCELLANEOUS PLOTS

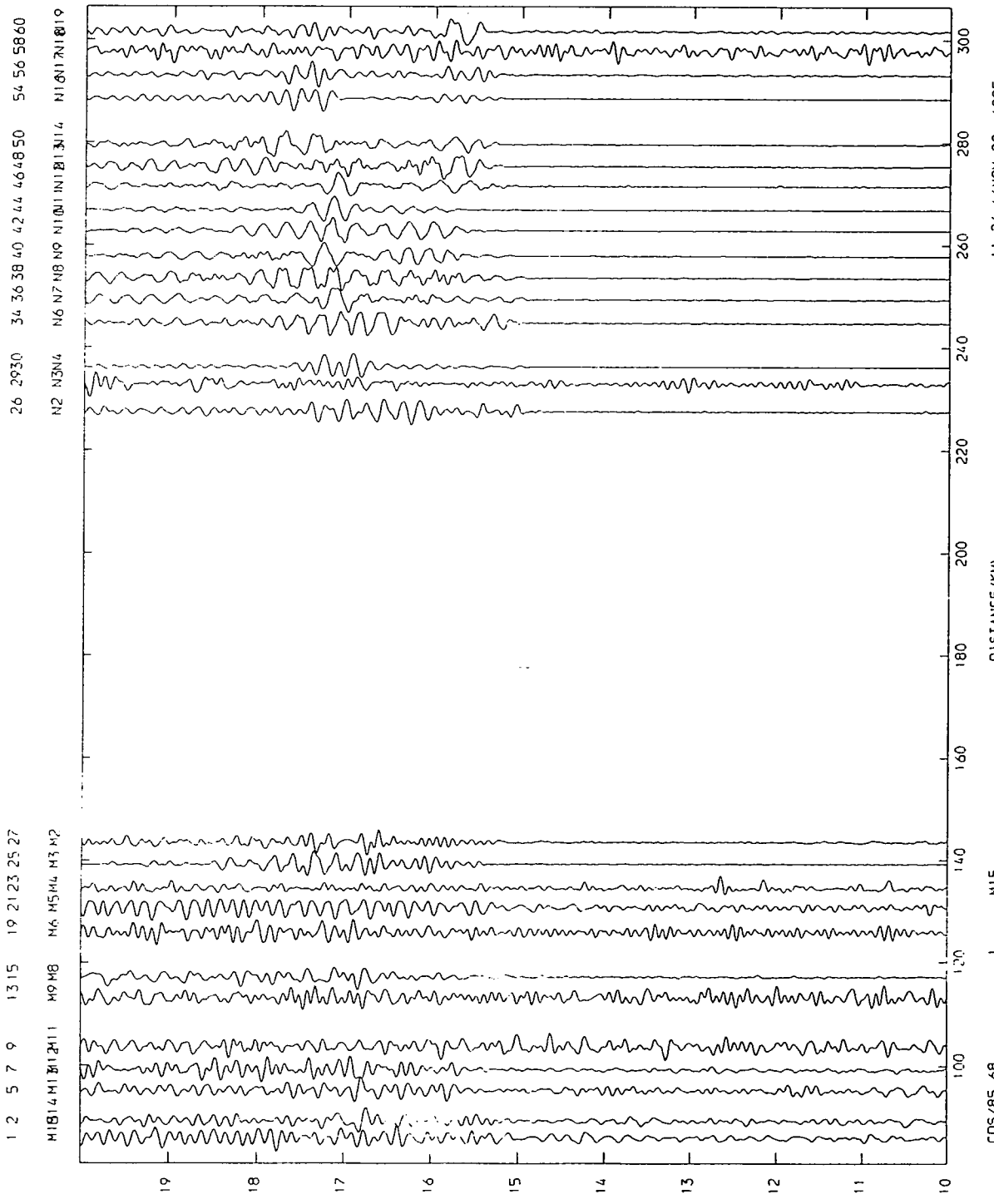
The plots provided in this appendix are all labelled with the plotting parametres and the name of the section plotted. In summary they are:

1. Common distance sections
2. Common mid-point sections
3. Common station plot for station R5 at the Eskdalemuir Array for the Irish Sea shots with amplitudes corrected to common gain and equalised
4. Travel-time regression plots for all common stations and all common shots.
5. Selection of travel-time plots showing the regression fit for some common stations and some common shots.

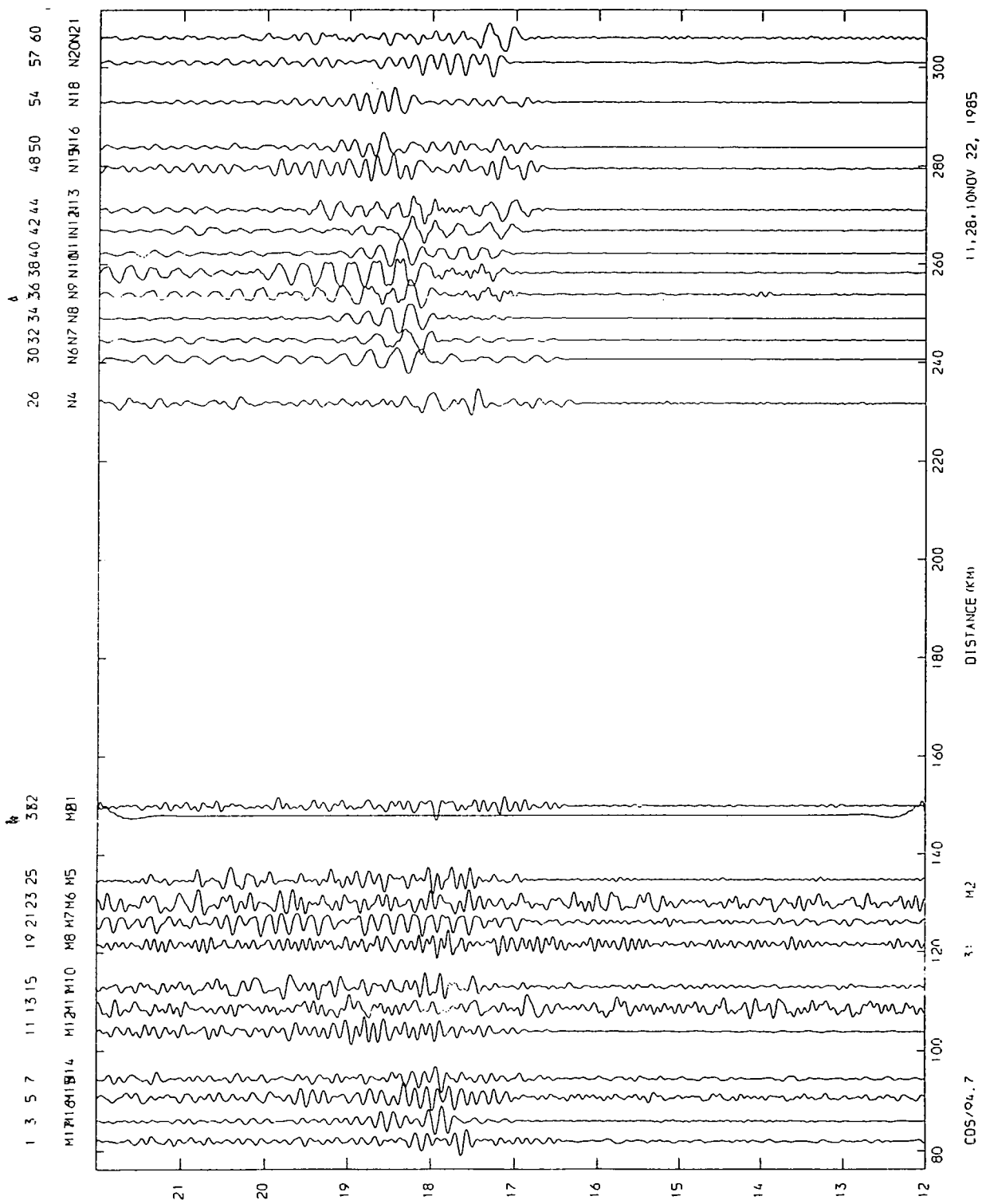
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



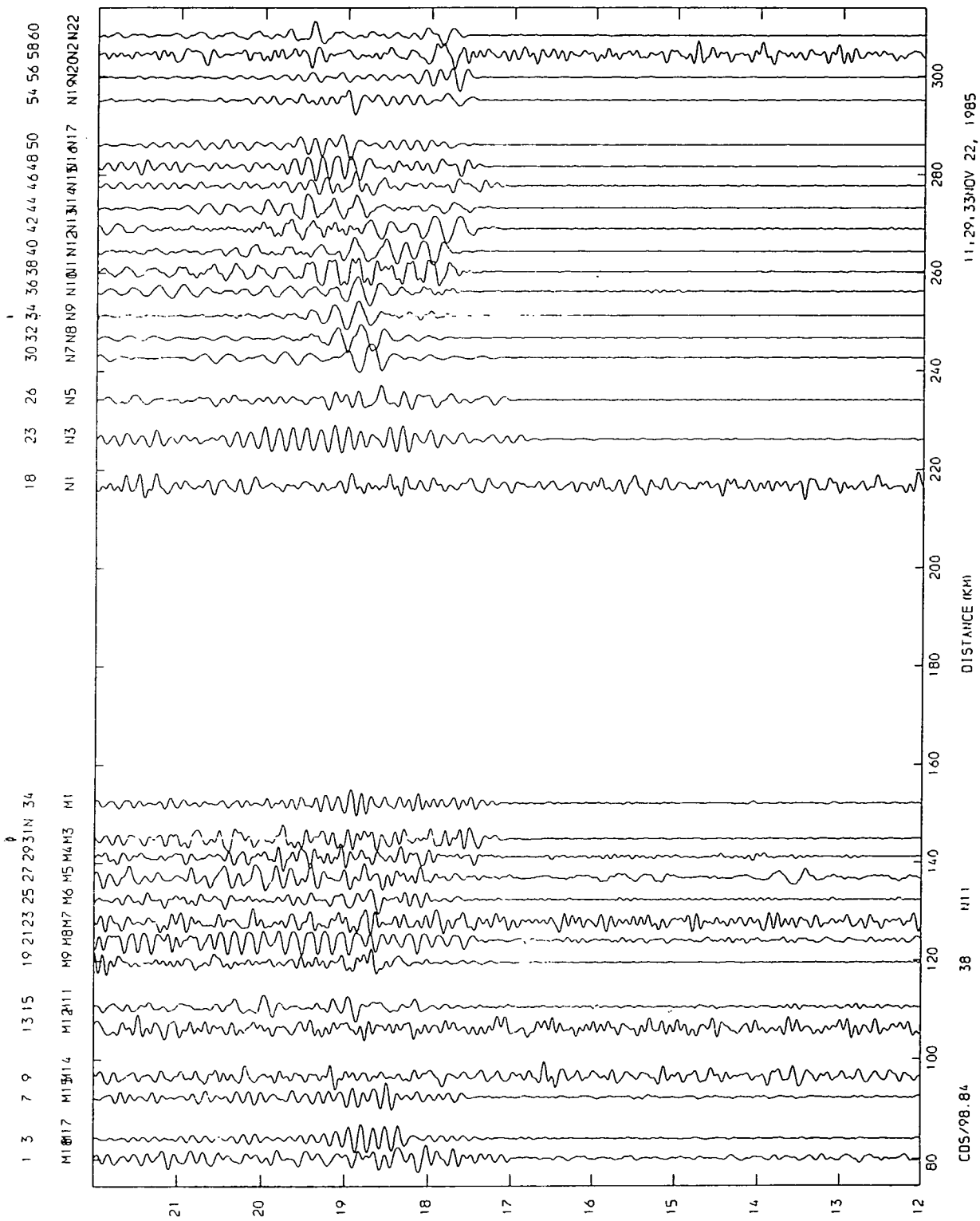
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



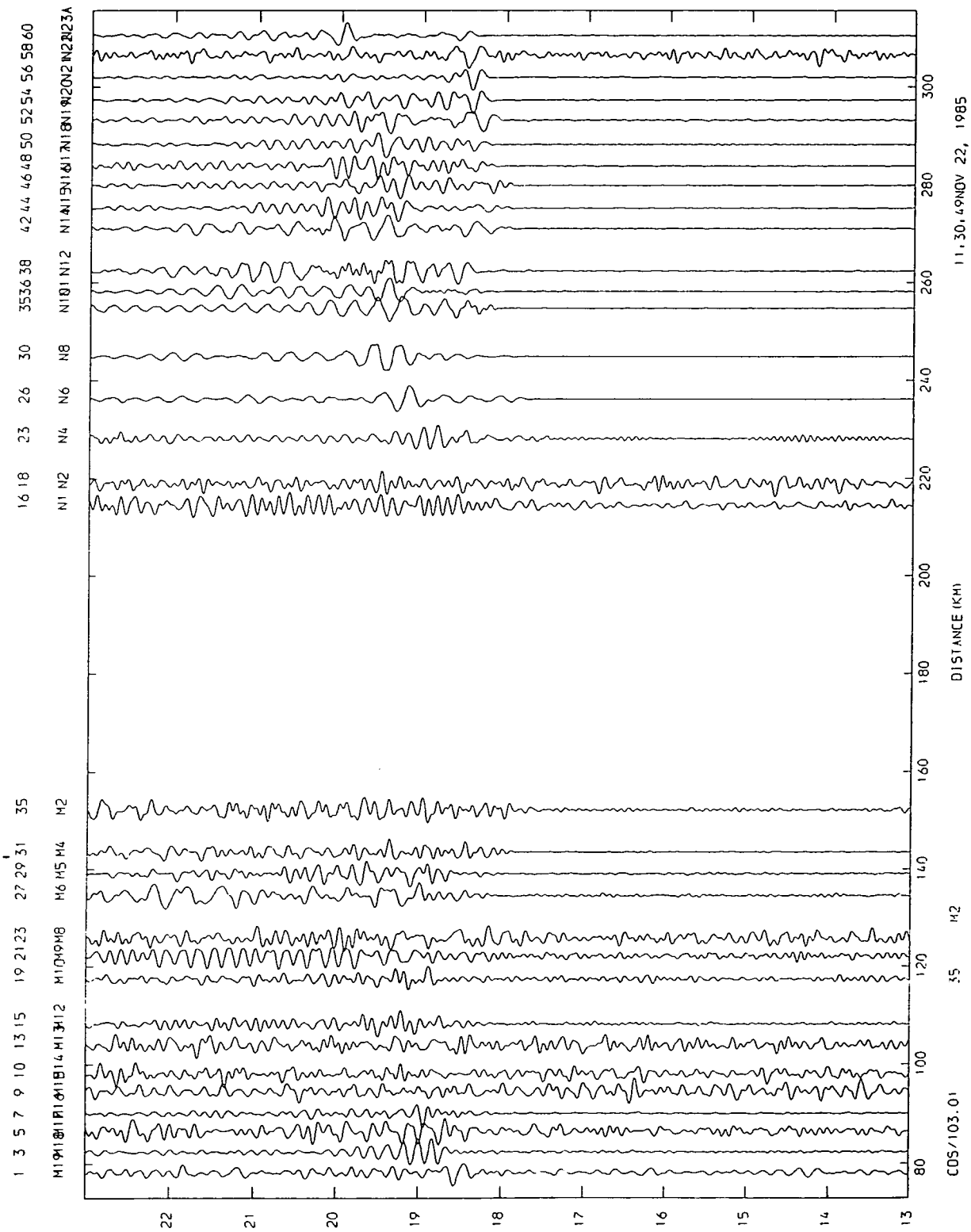
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



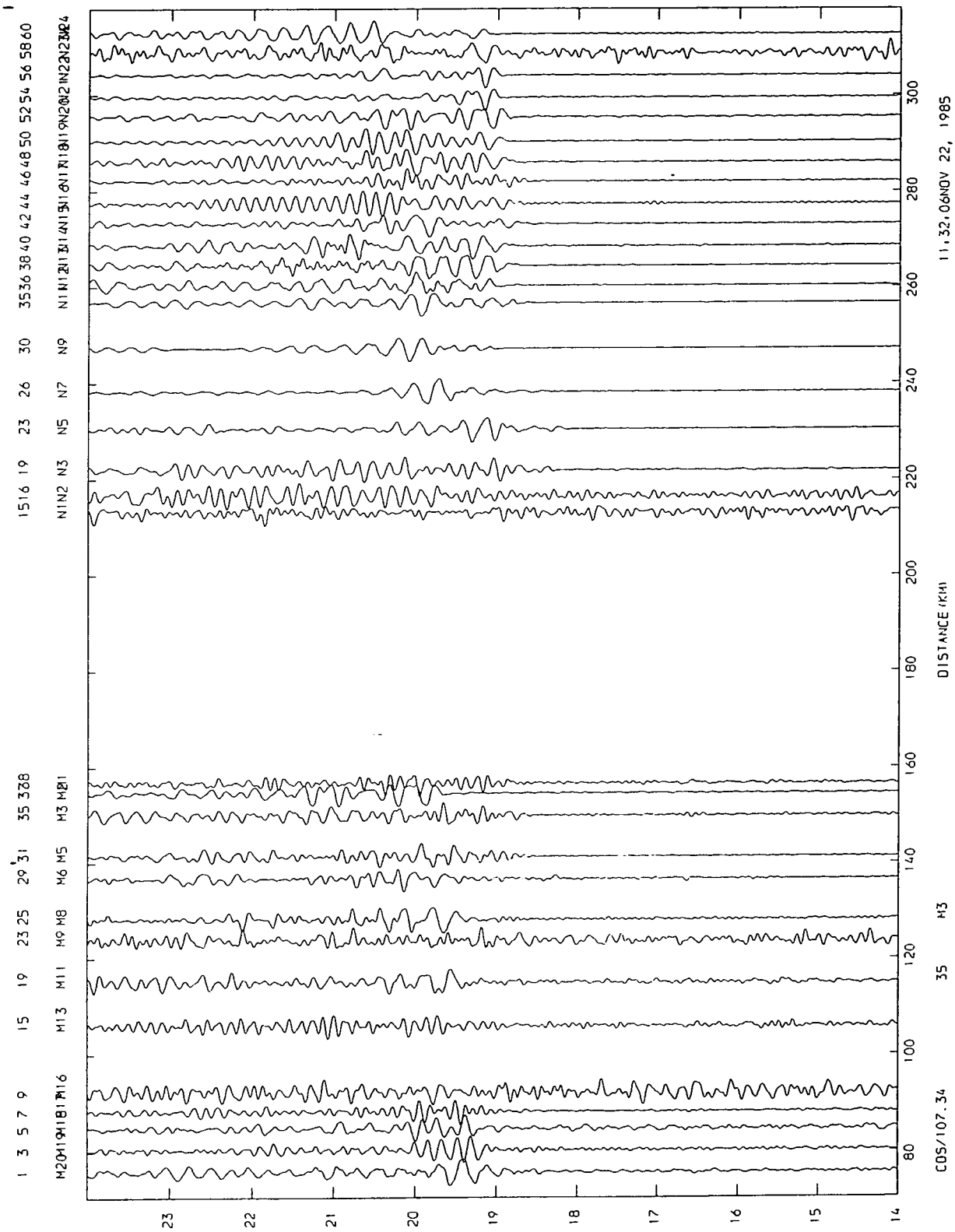
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



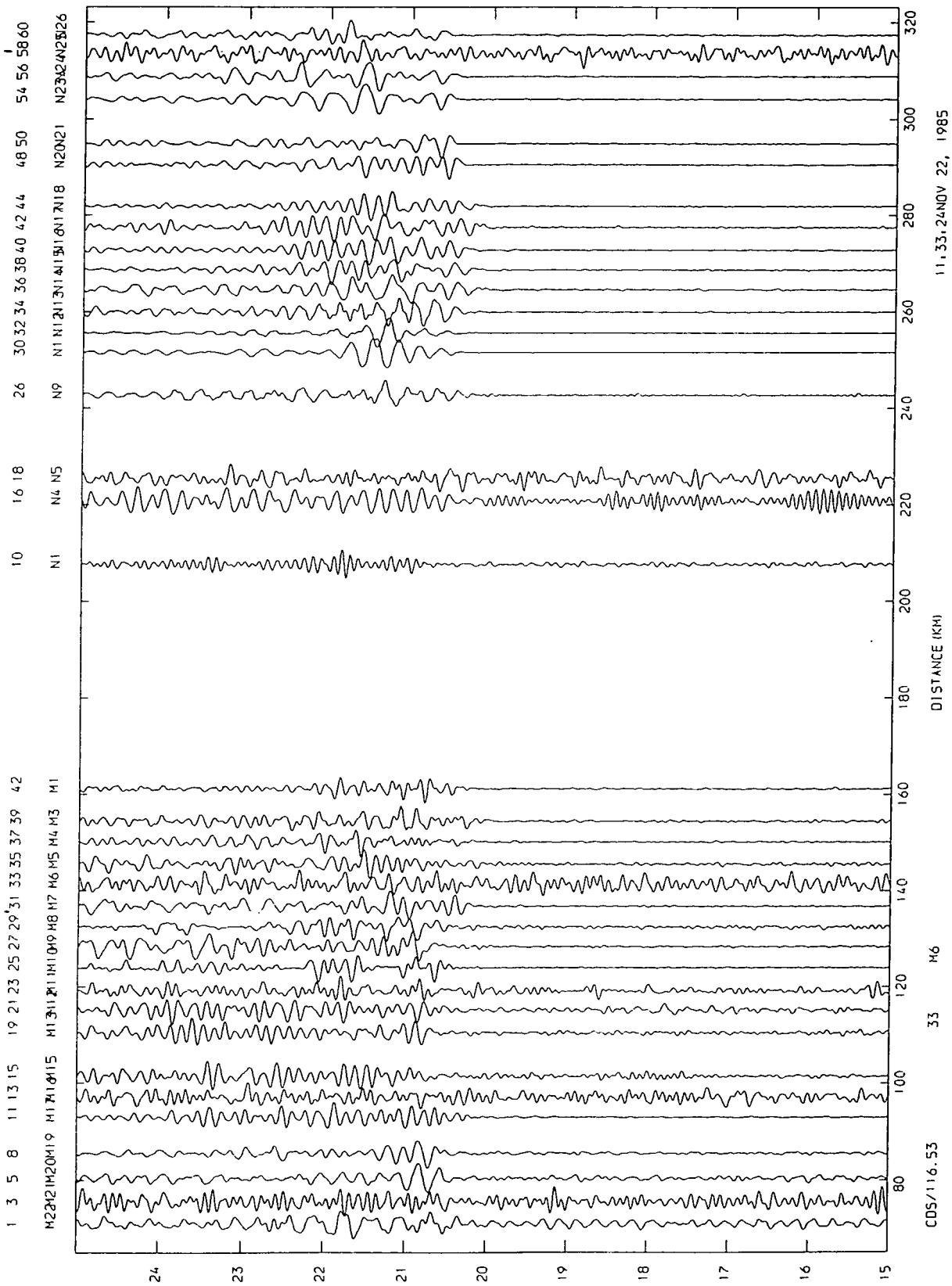
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



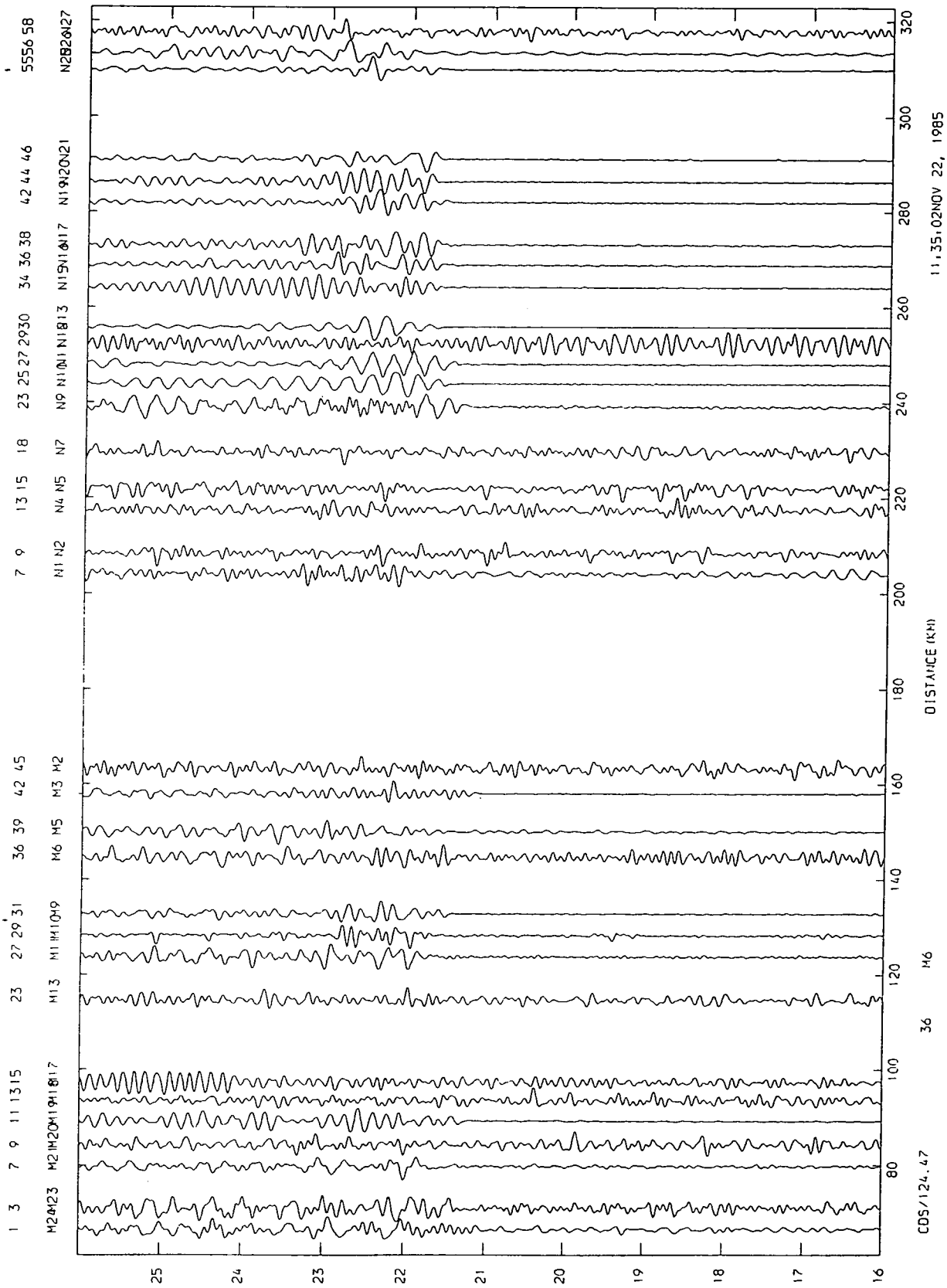
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



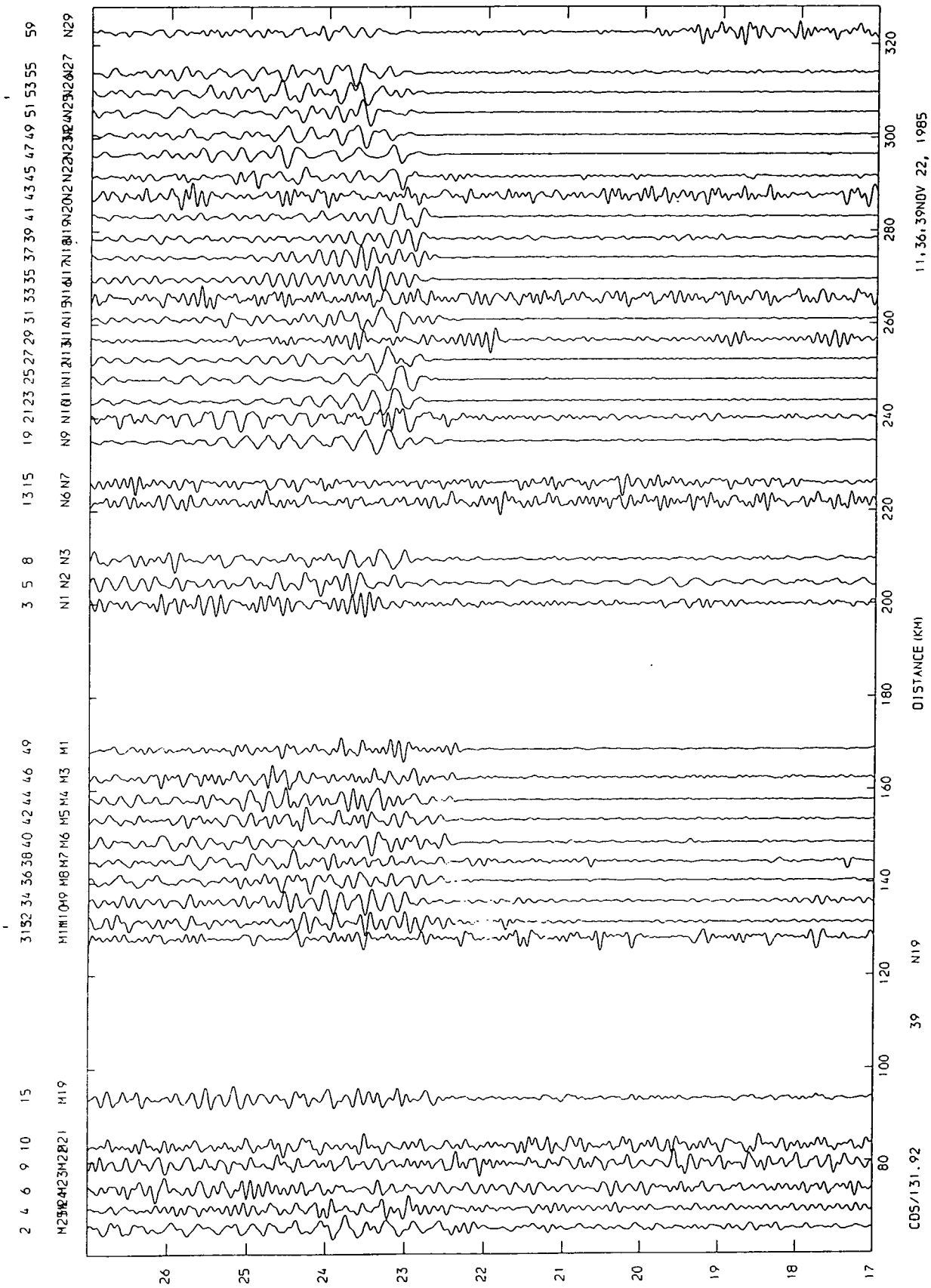
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



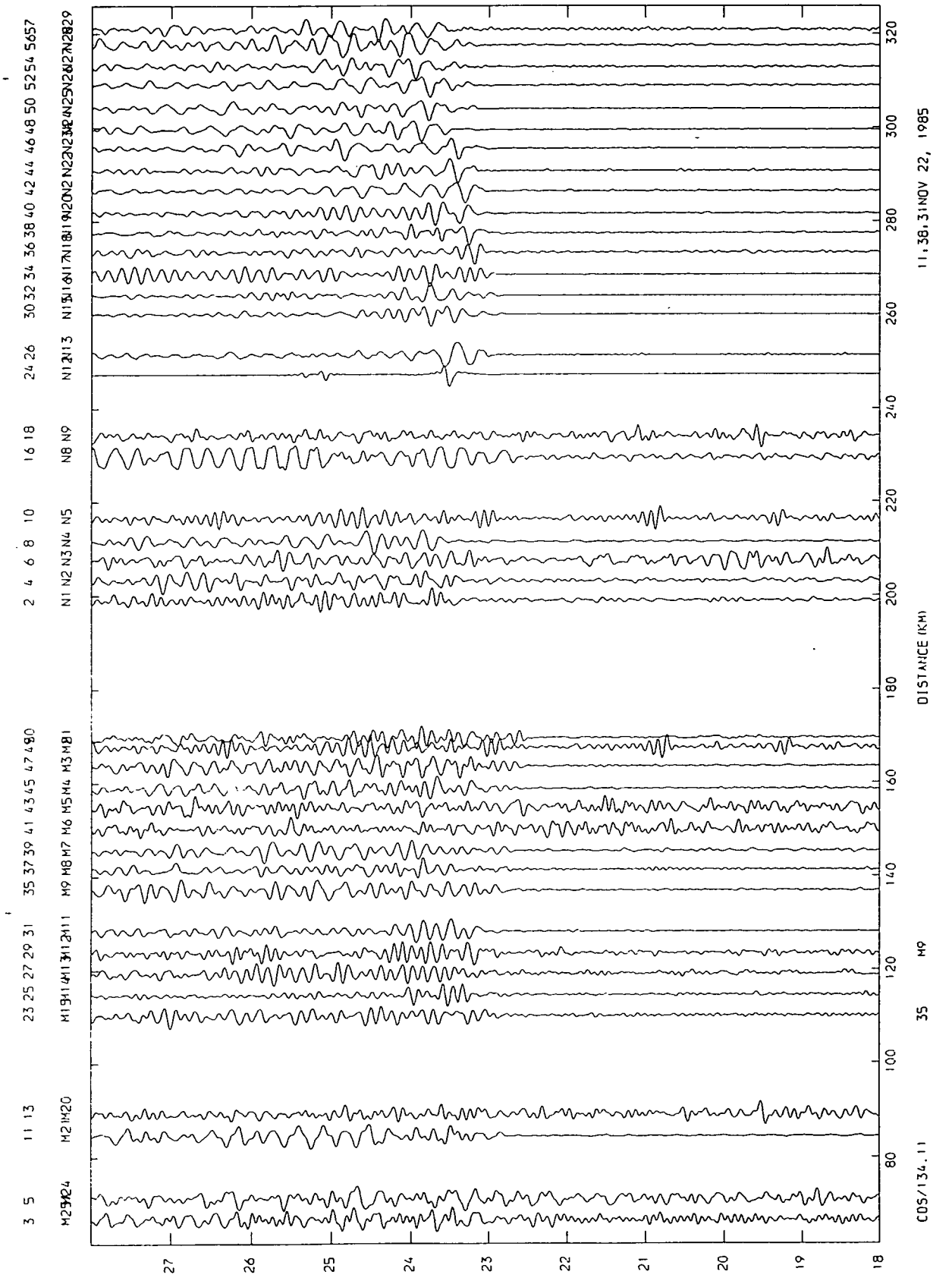
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



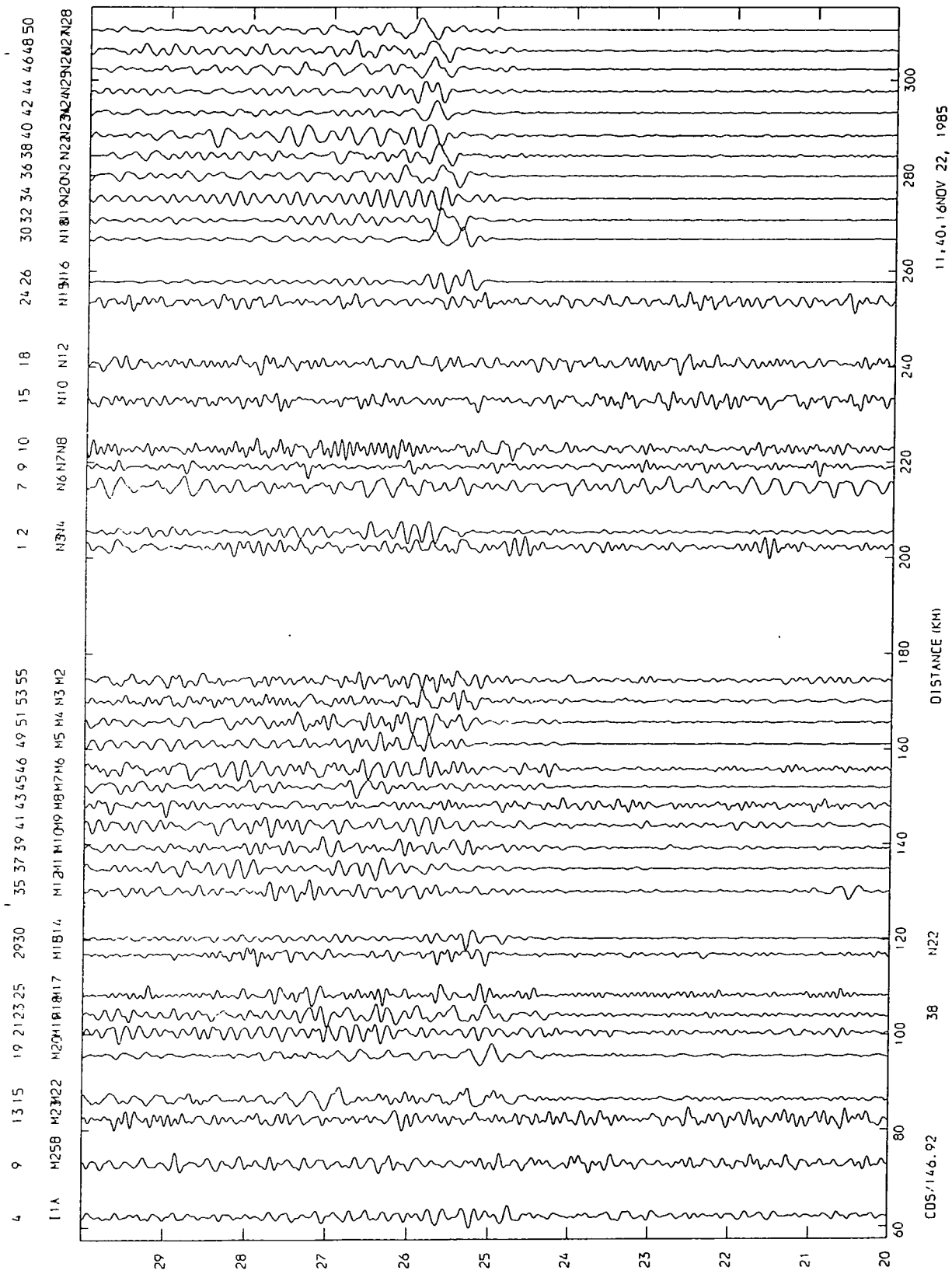
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



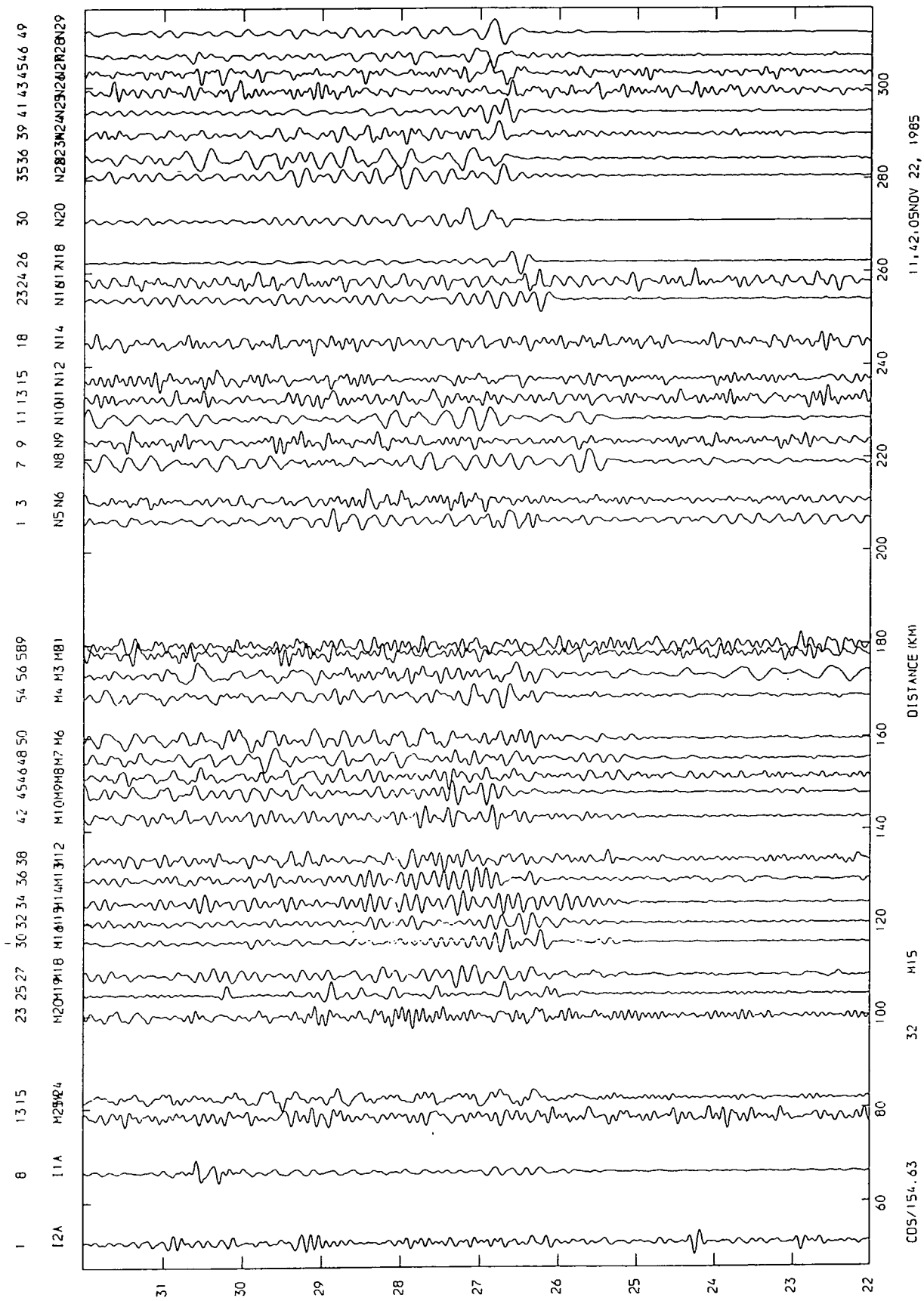
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



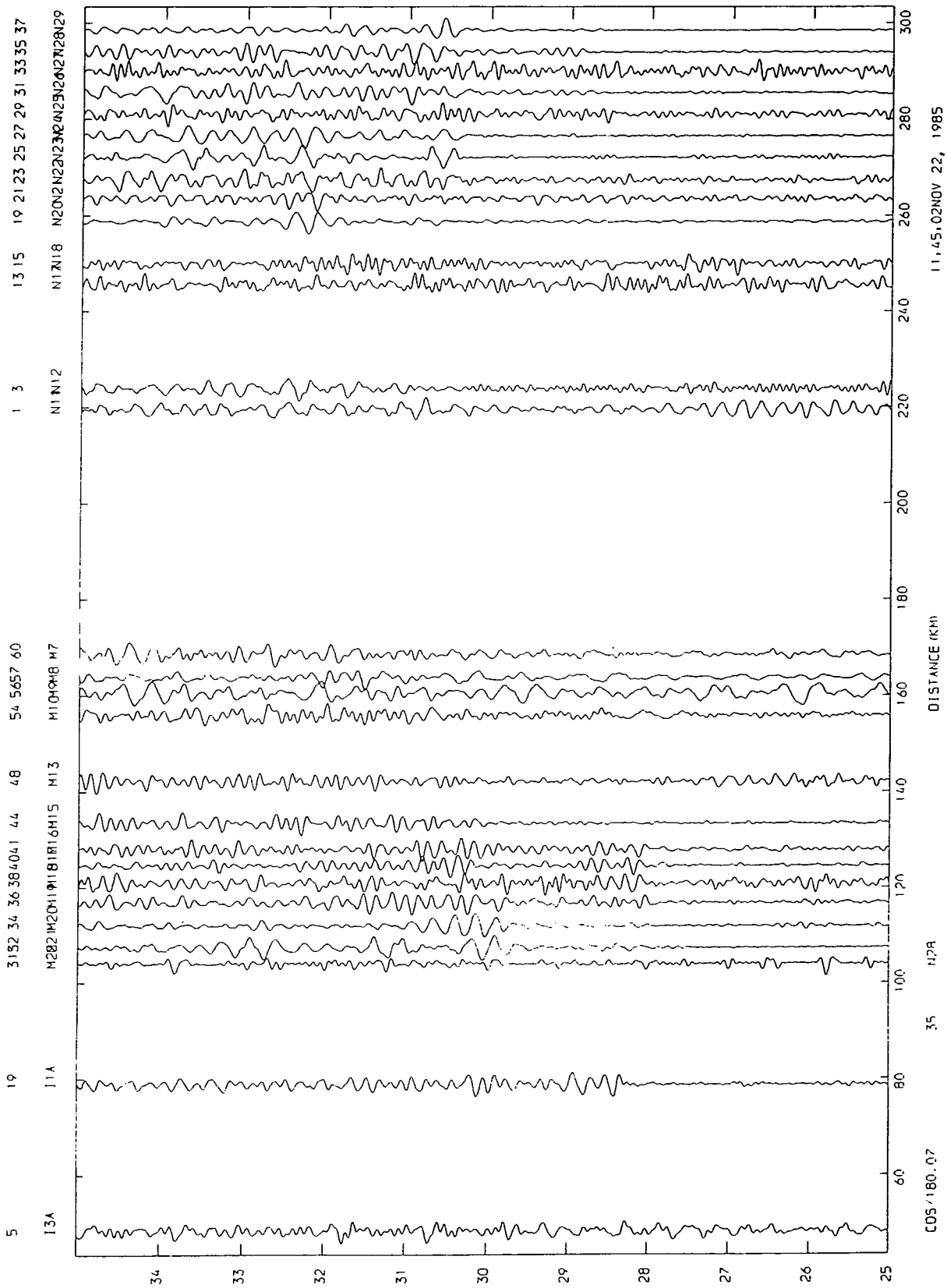
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



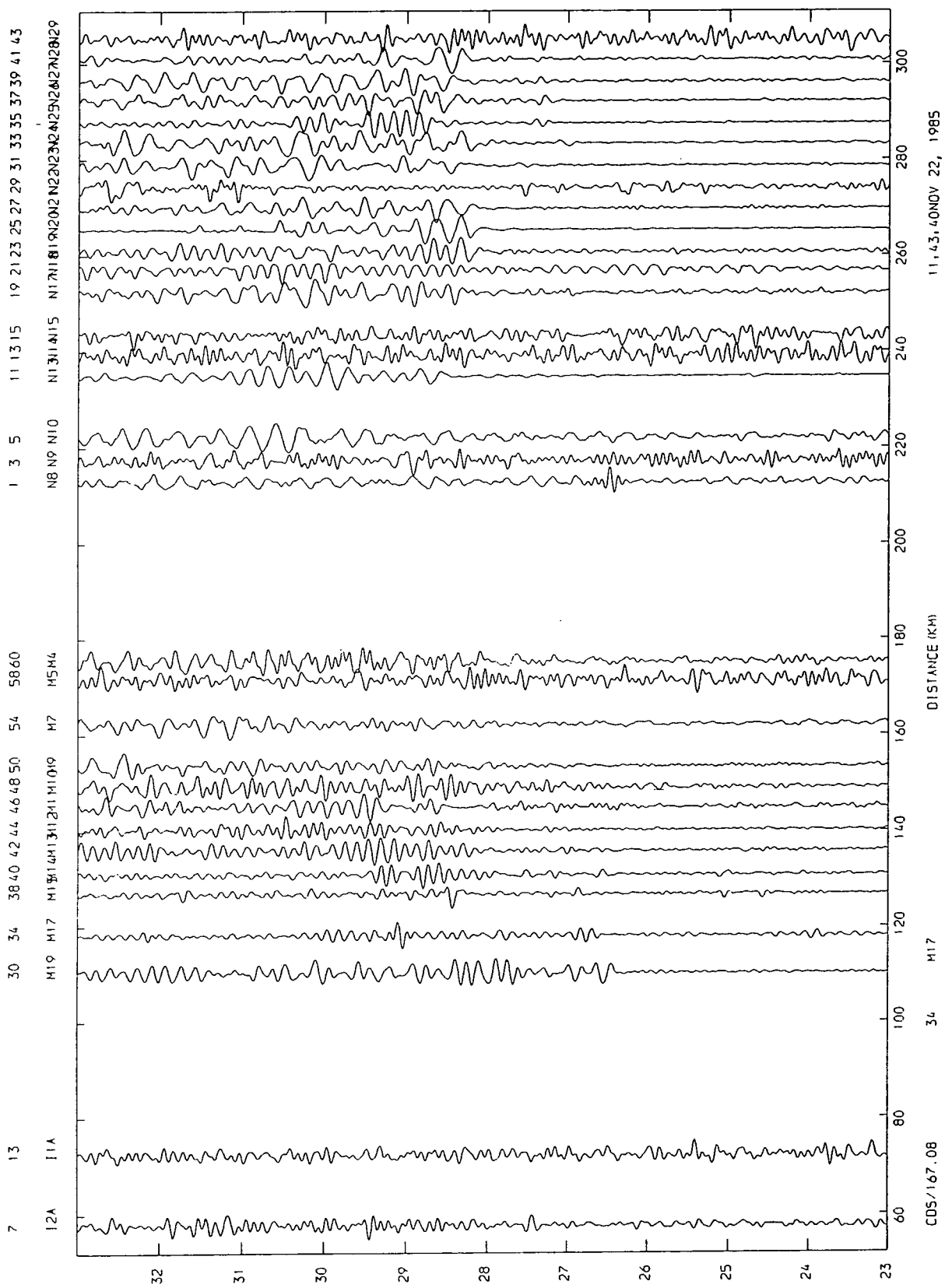
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



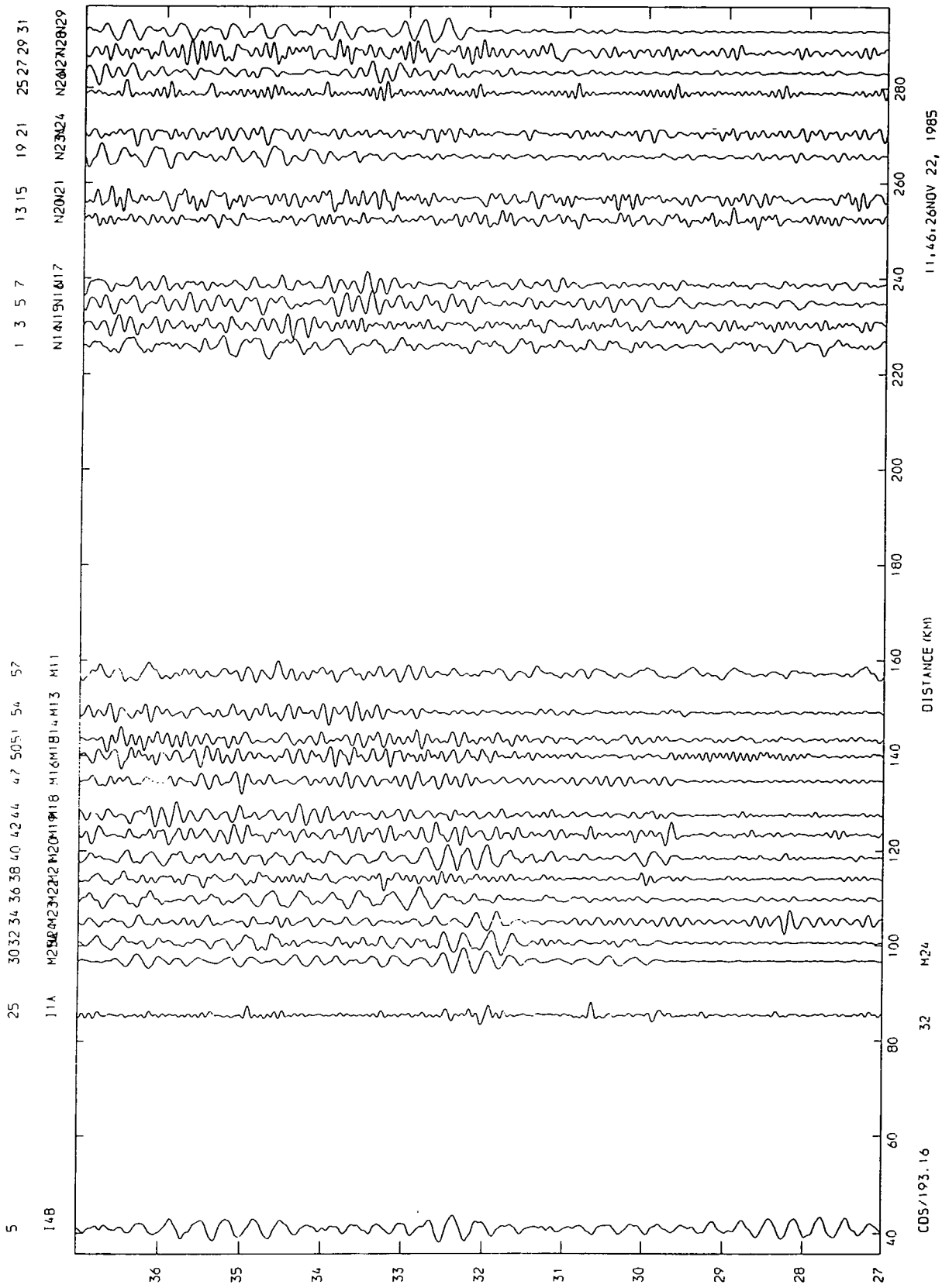
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



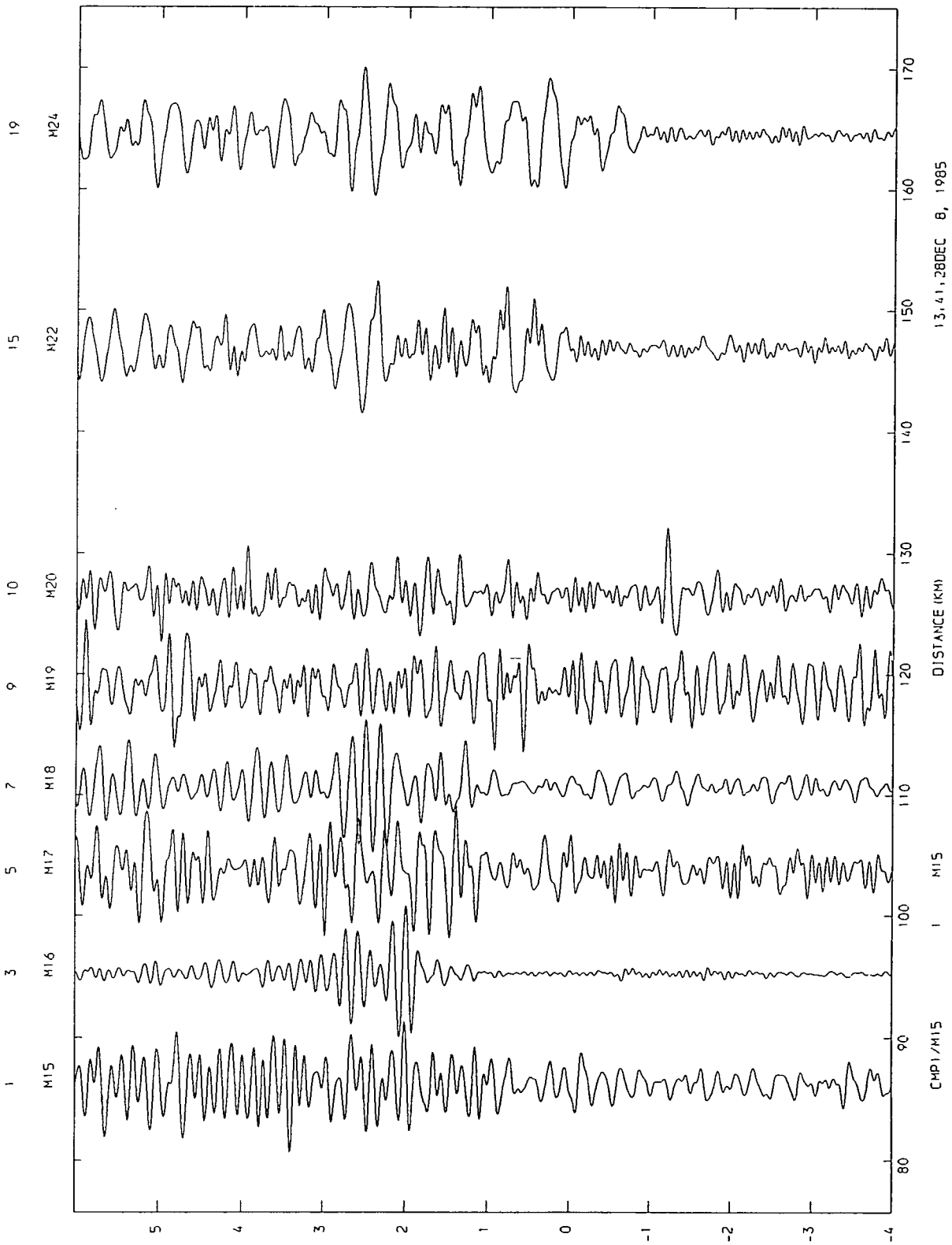
FILTERED 2.0 12.0 TIME (SECS) RED TO 0.0 KM/SEC EQUALIZED



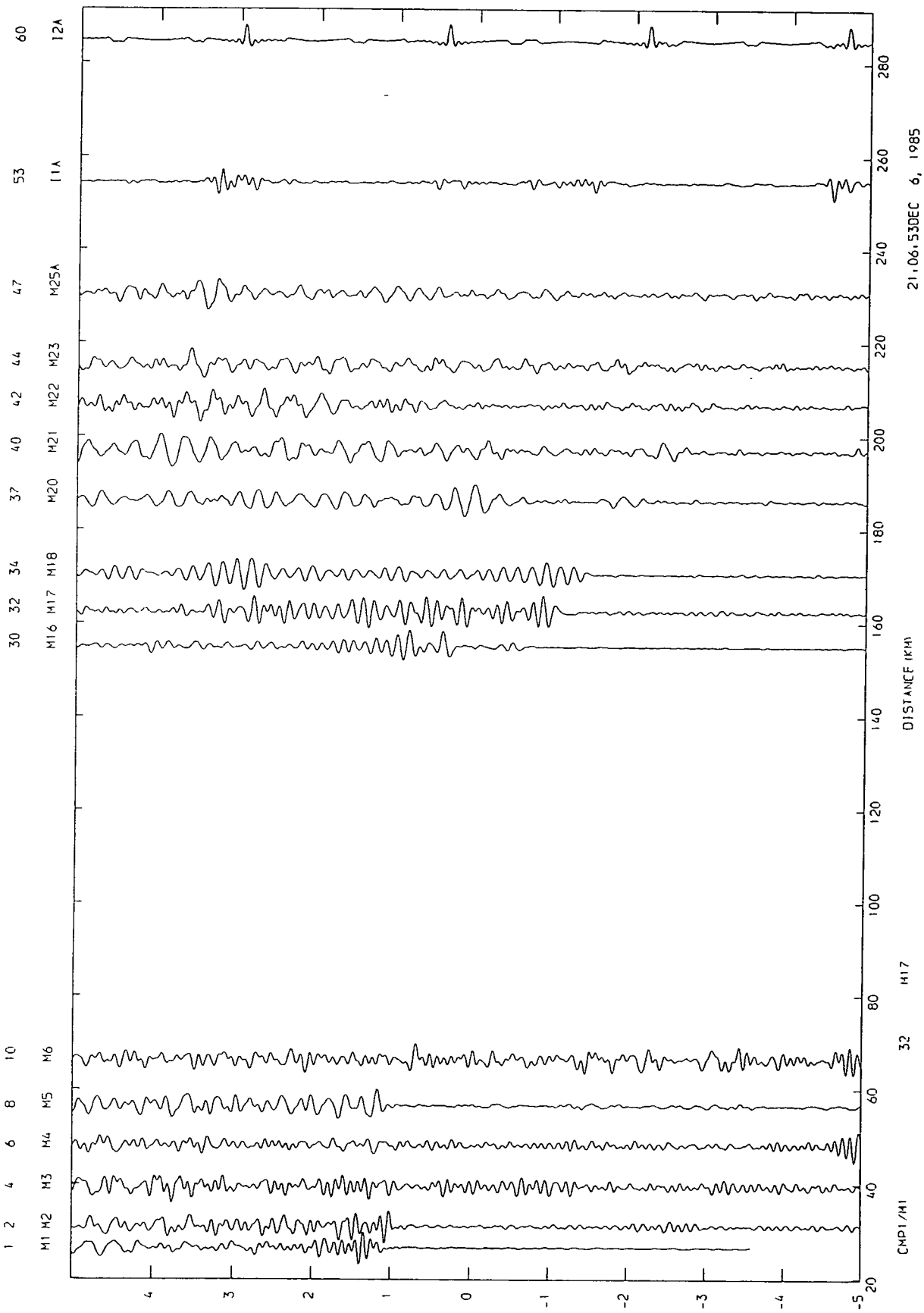
FILTERED 2.0 12.0 TIME(SECS) RED TO 0.0 KM/SEC EQUALIZED



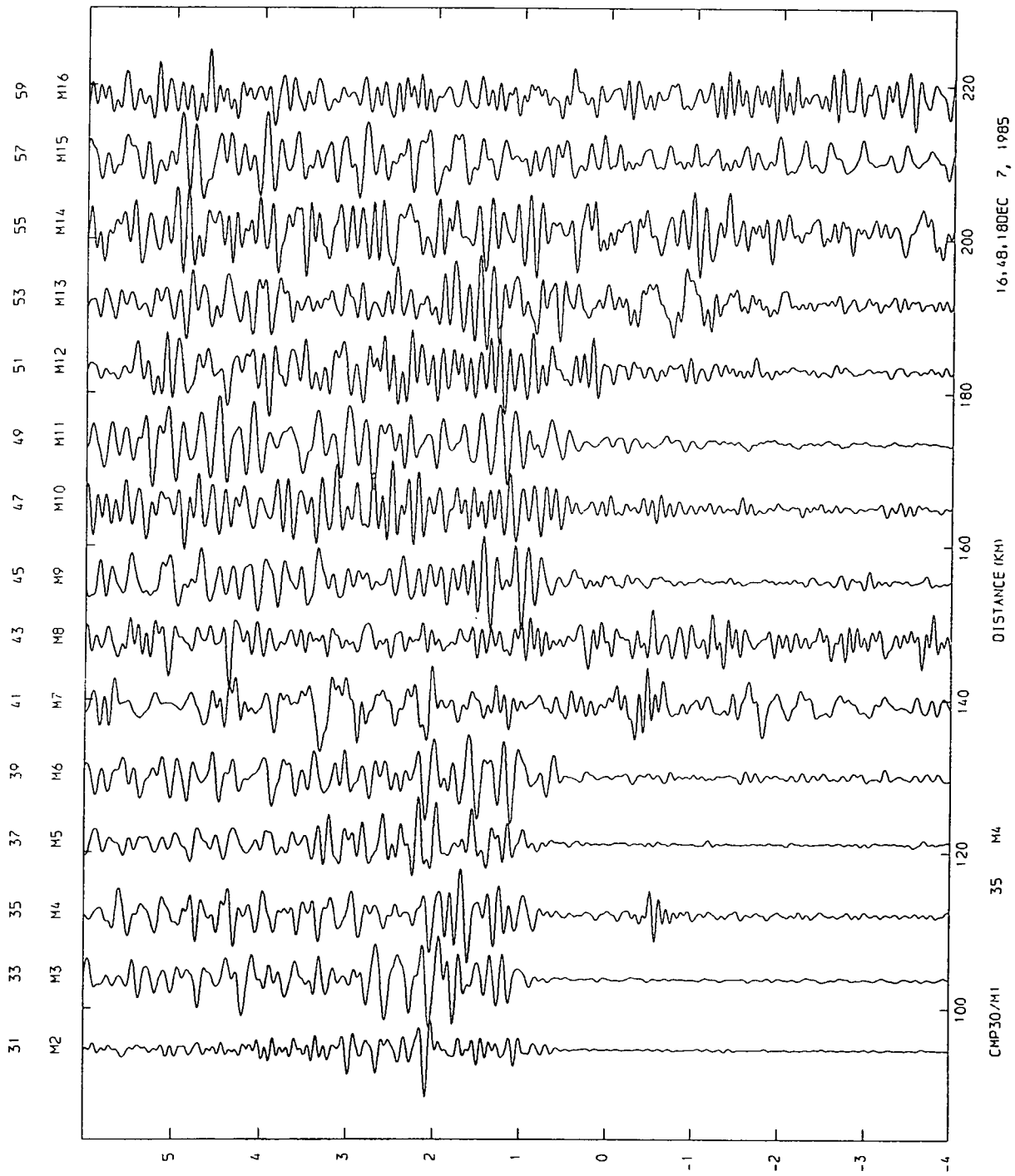
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



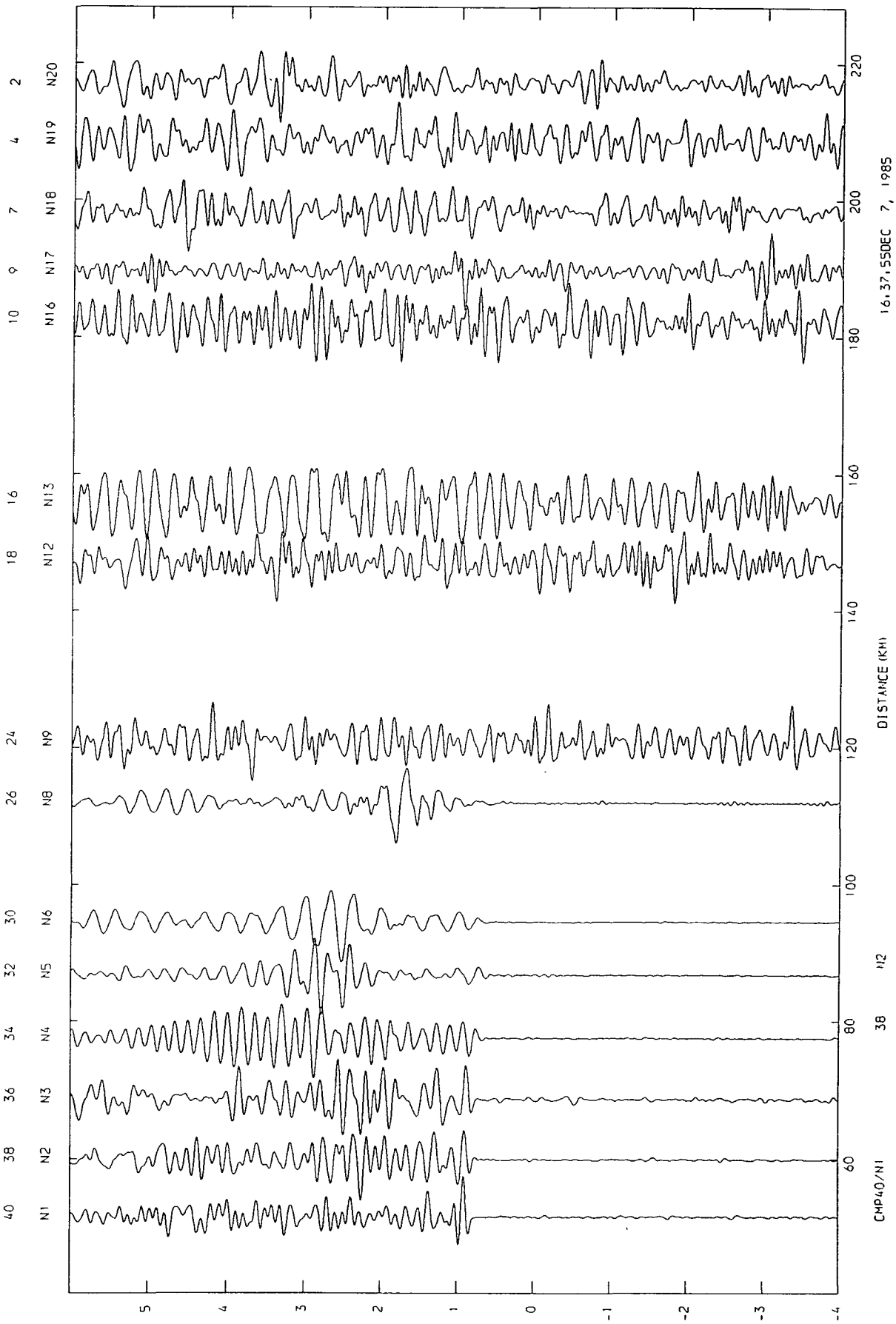
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



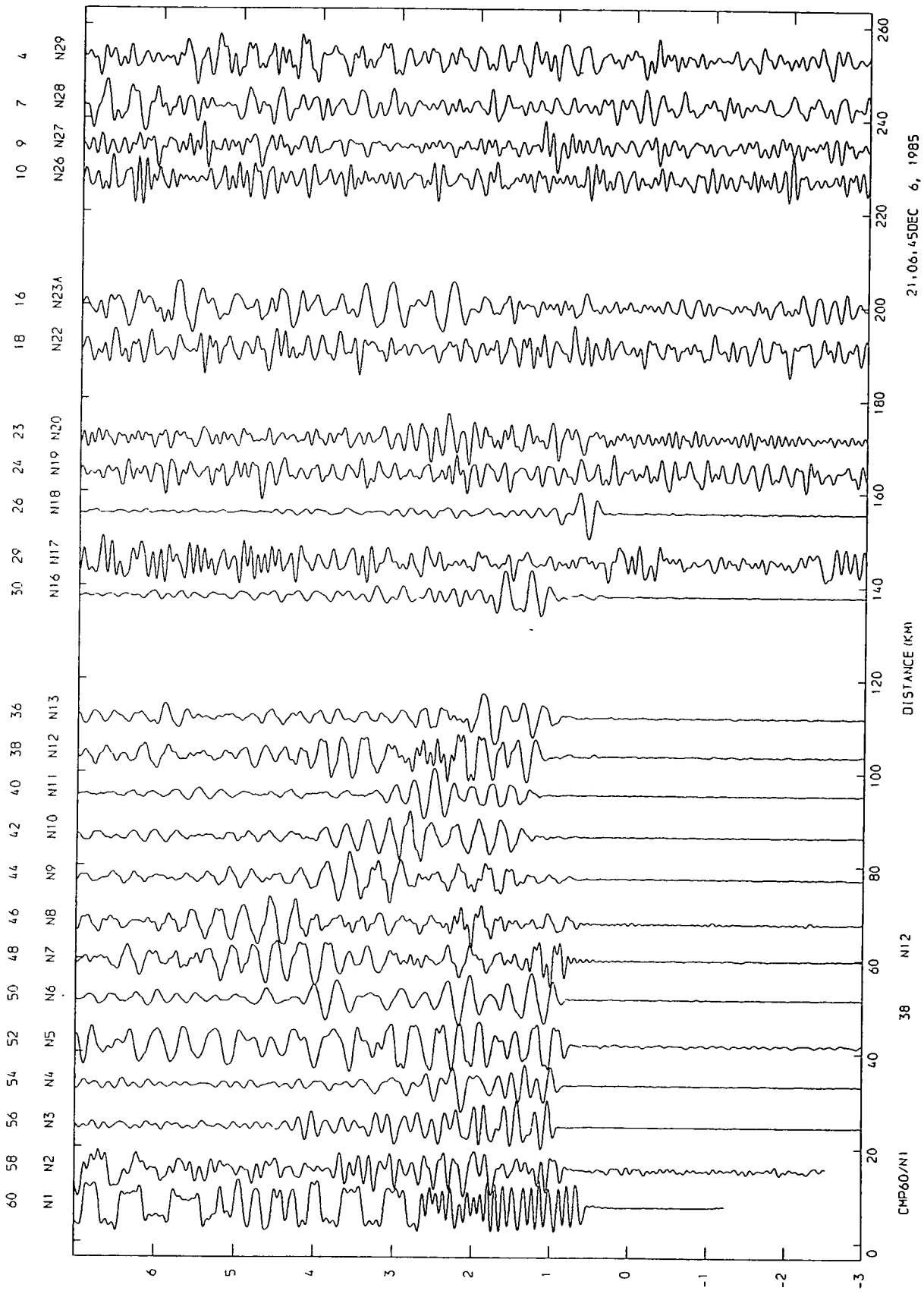
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



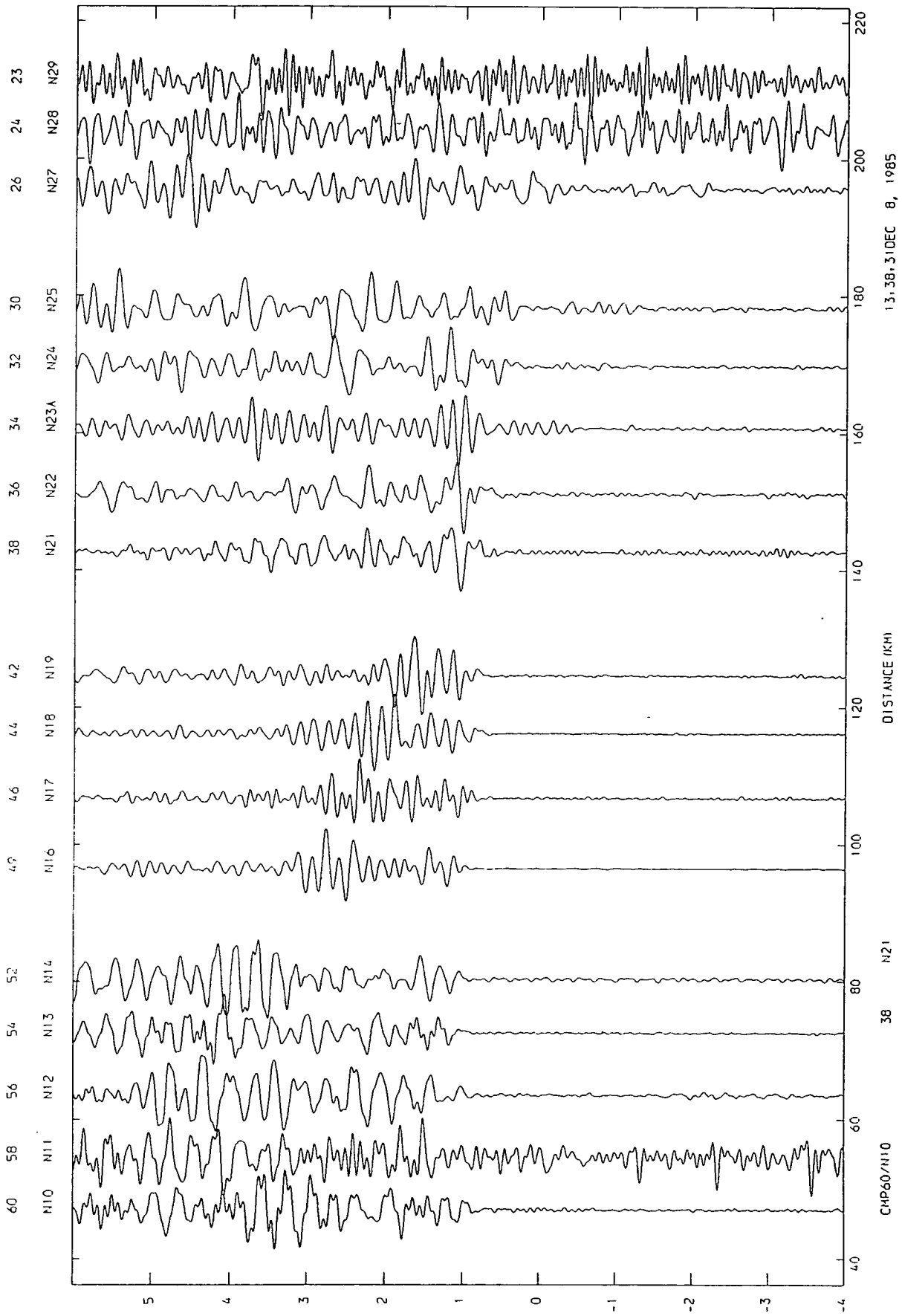
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED



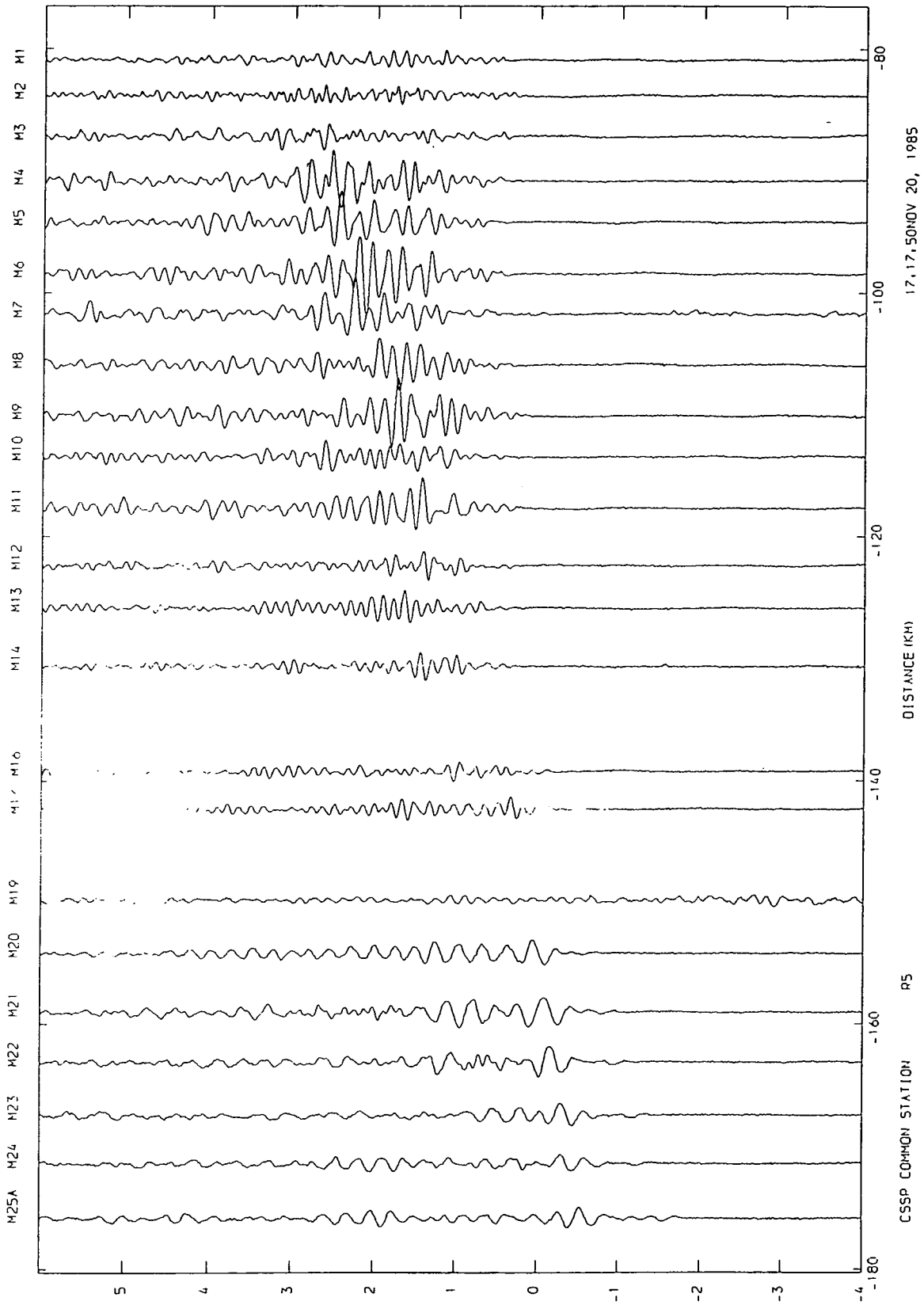
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED



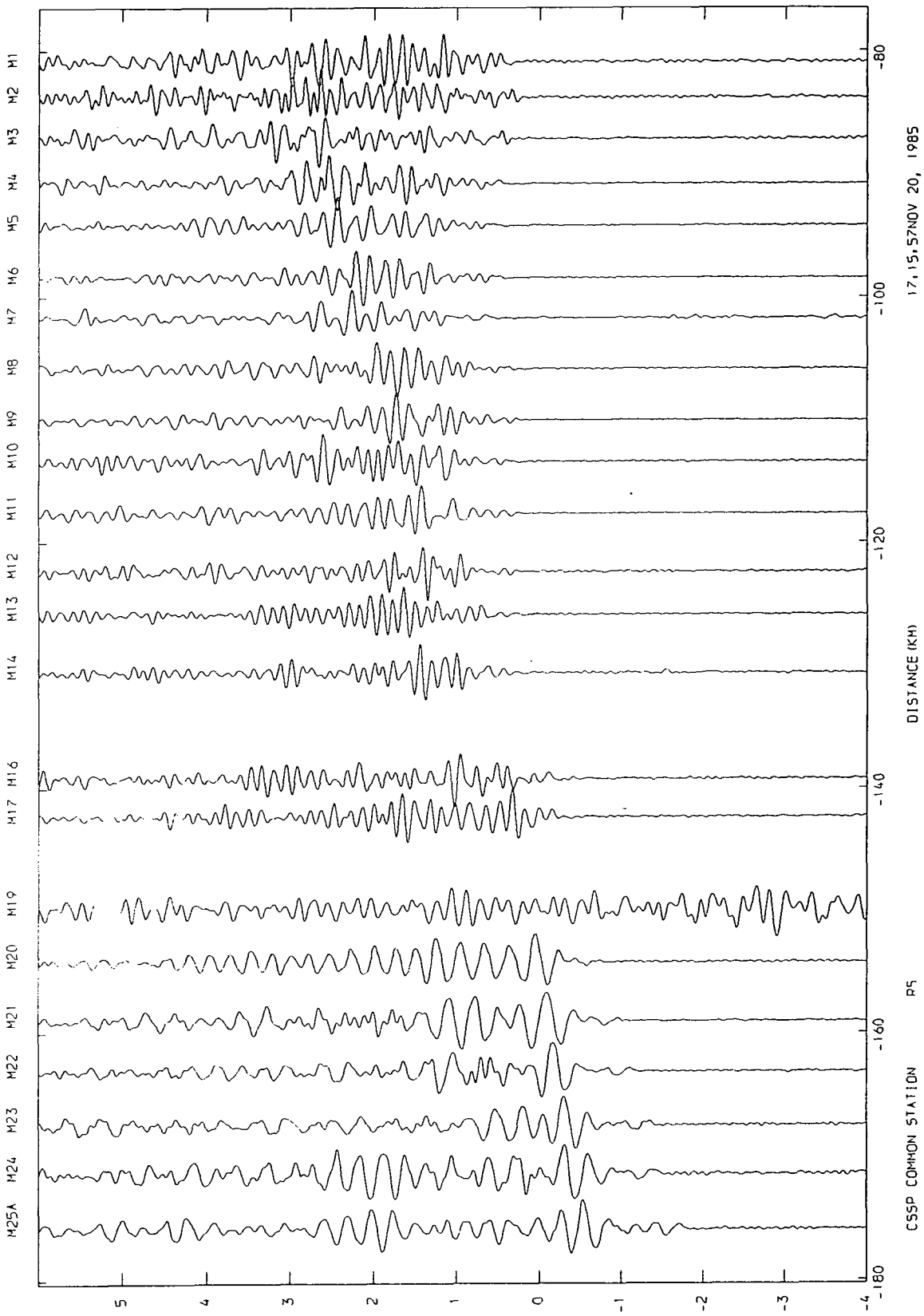
FILTERED 2.0 12.0 TIME (SECS) RED TO 6.0 KM/SEC EQUALIZED

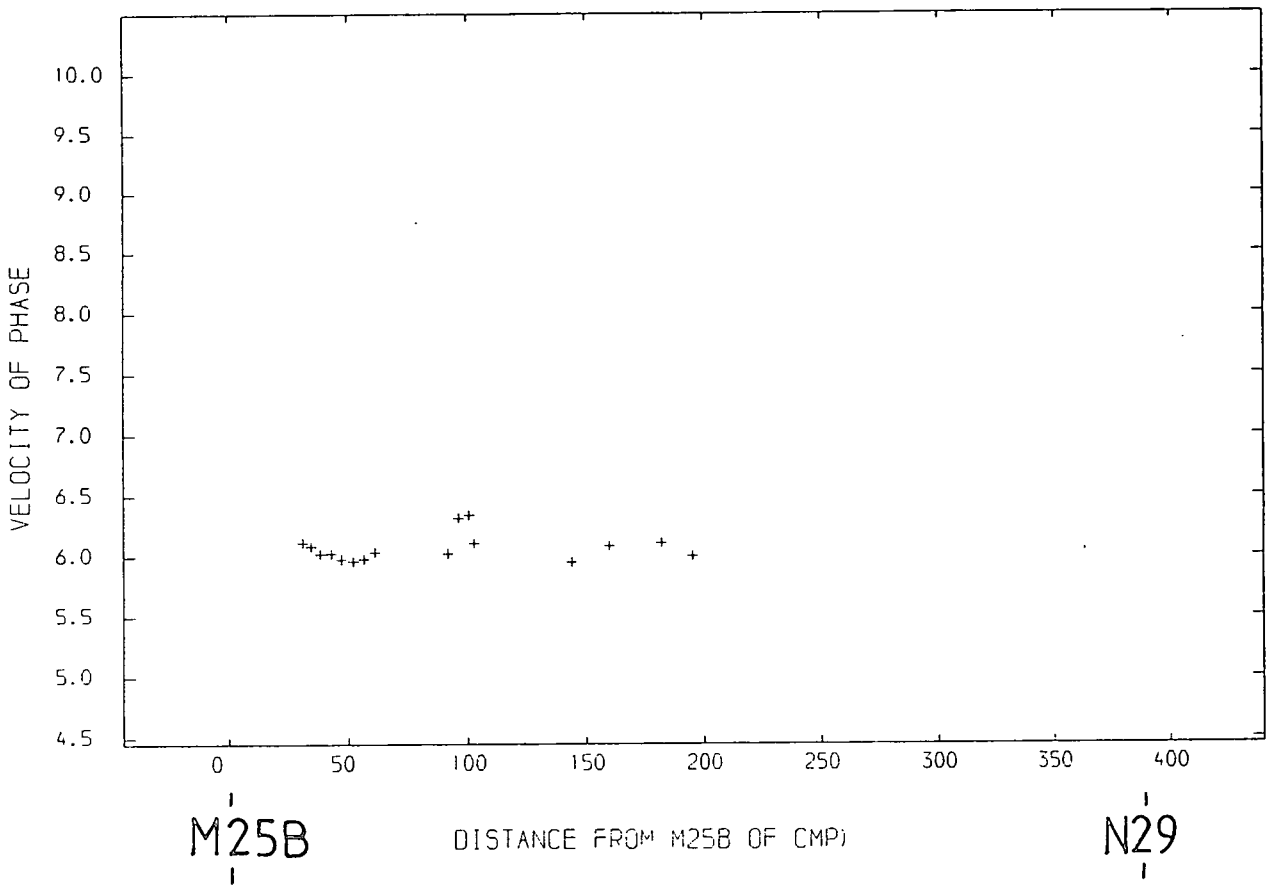


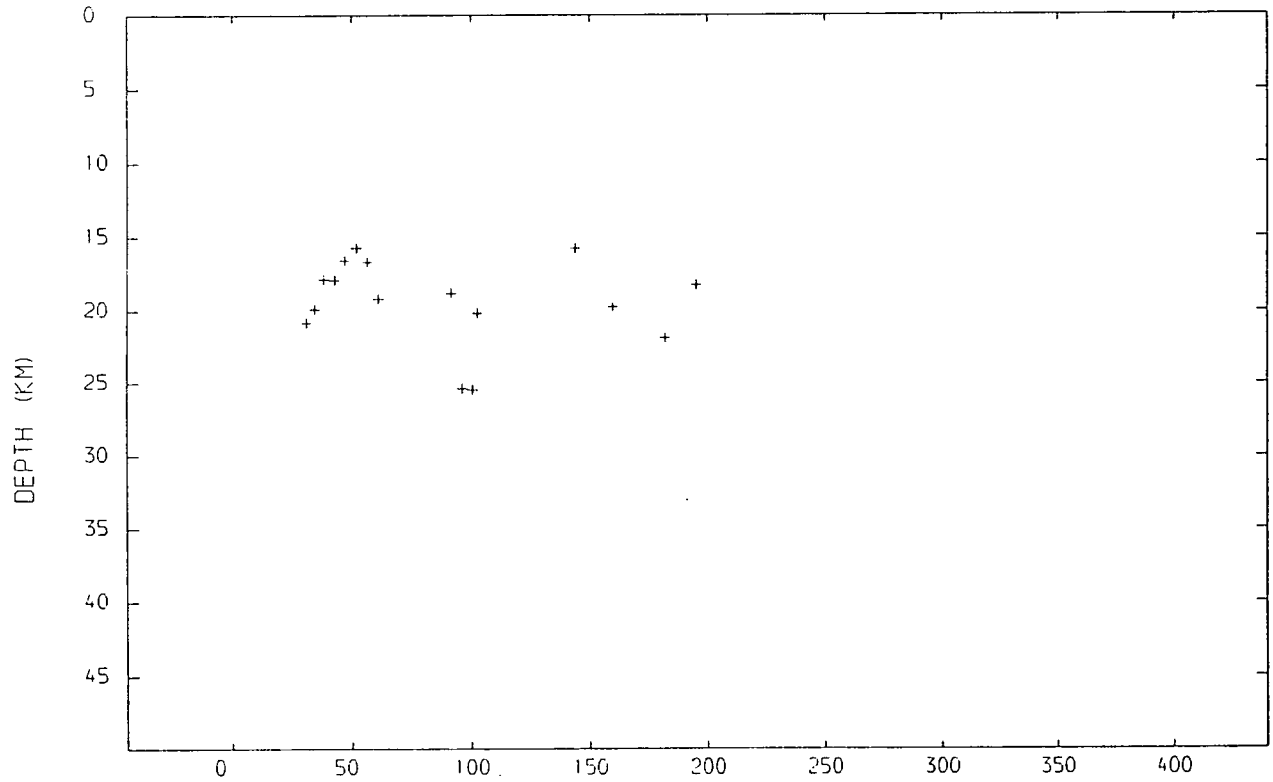
UNFILTERED TIME (SECS) RED TO 6.0 KM/SEC TRACE AMPFAC 0.0000037



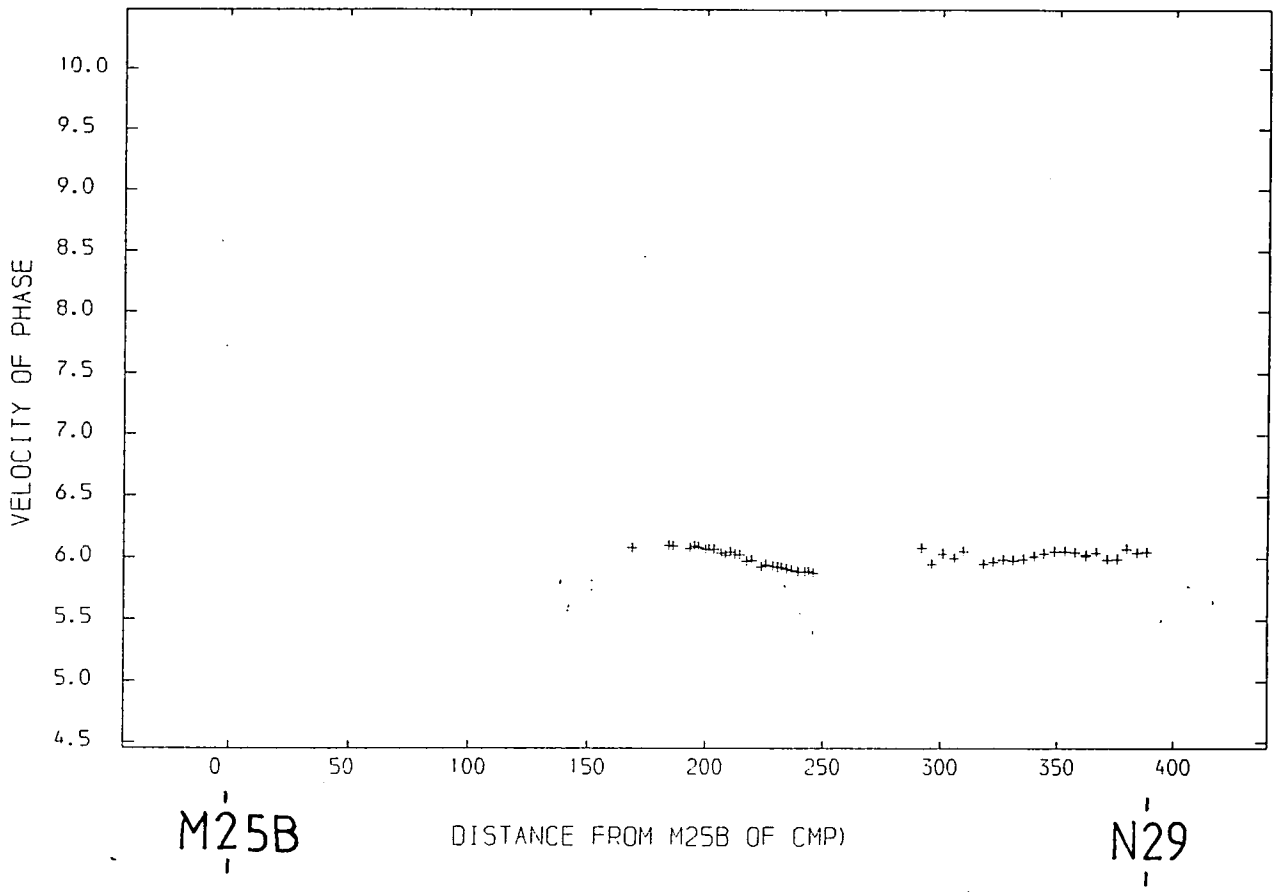
FILTERED 2.0 12.0 TIME(SECS) RED TO 6.0 KM/SEC EQUALIZED

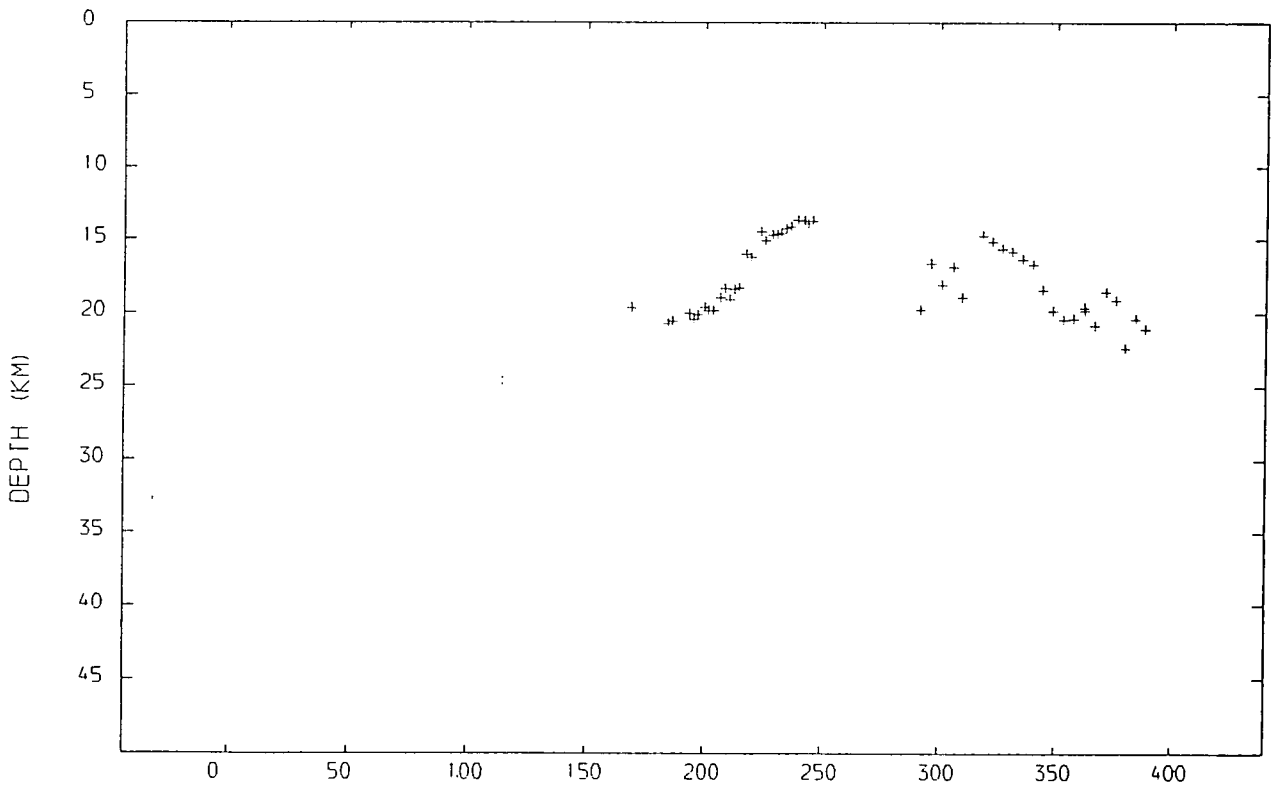




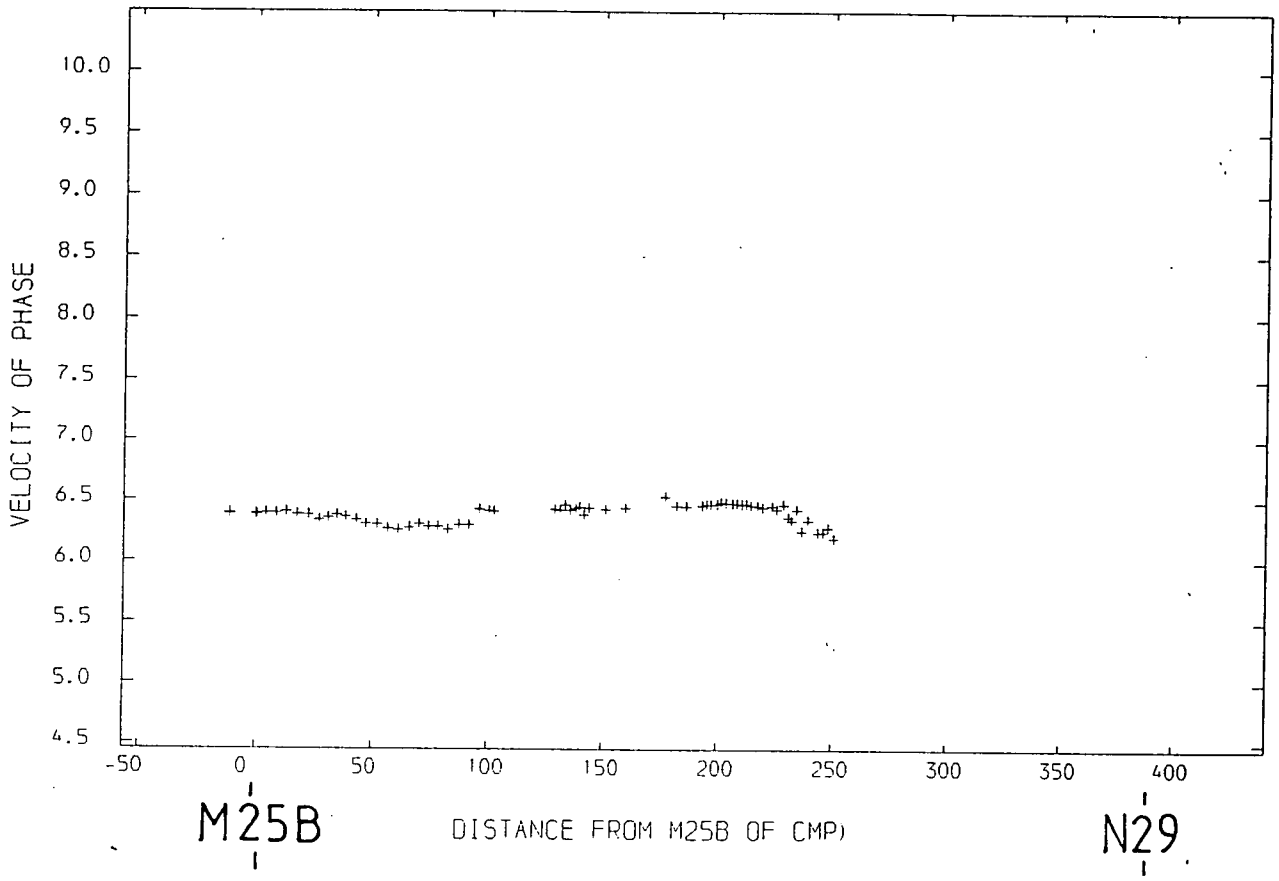


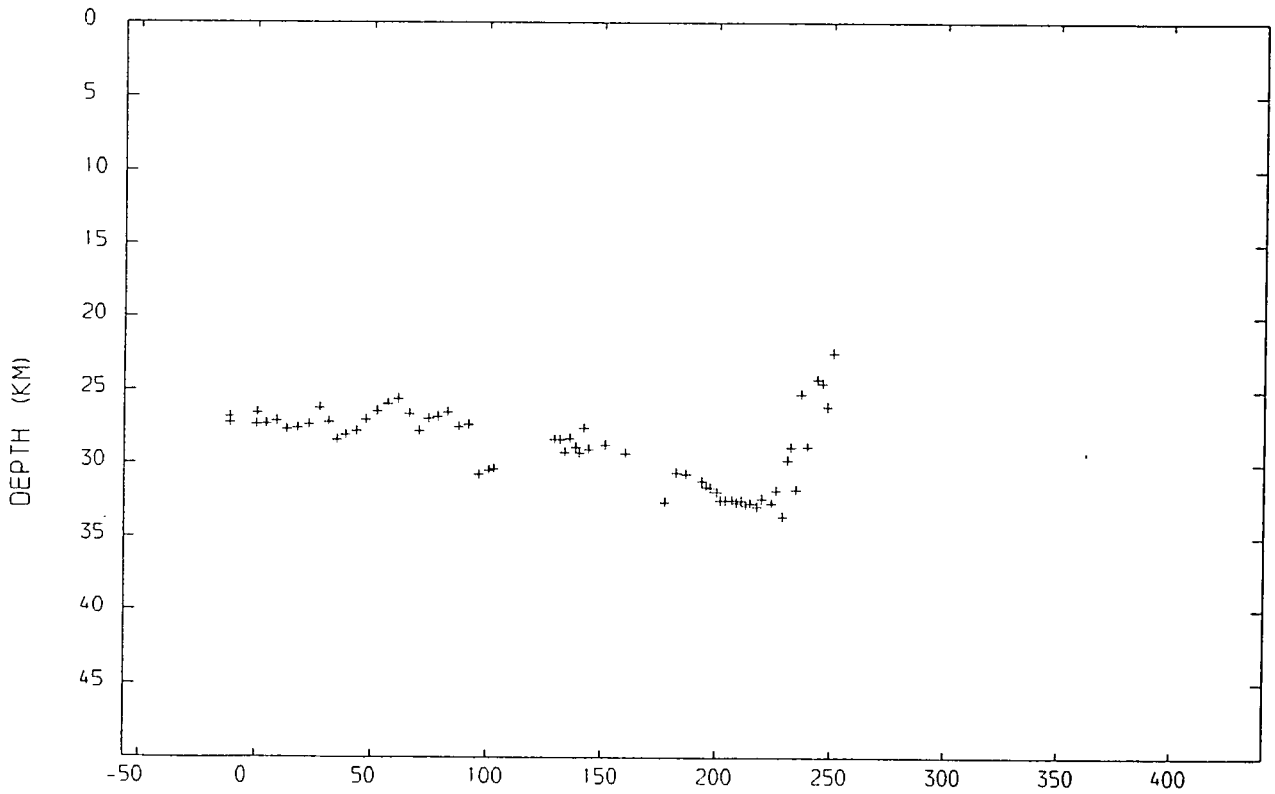
ISEA TT-PEG PESULTS PCP



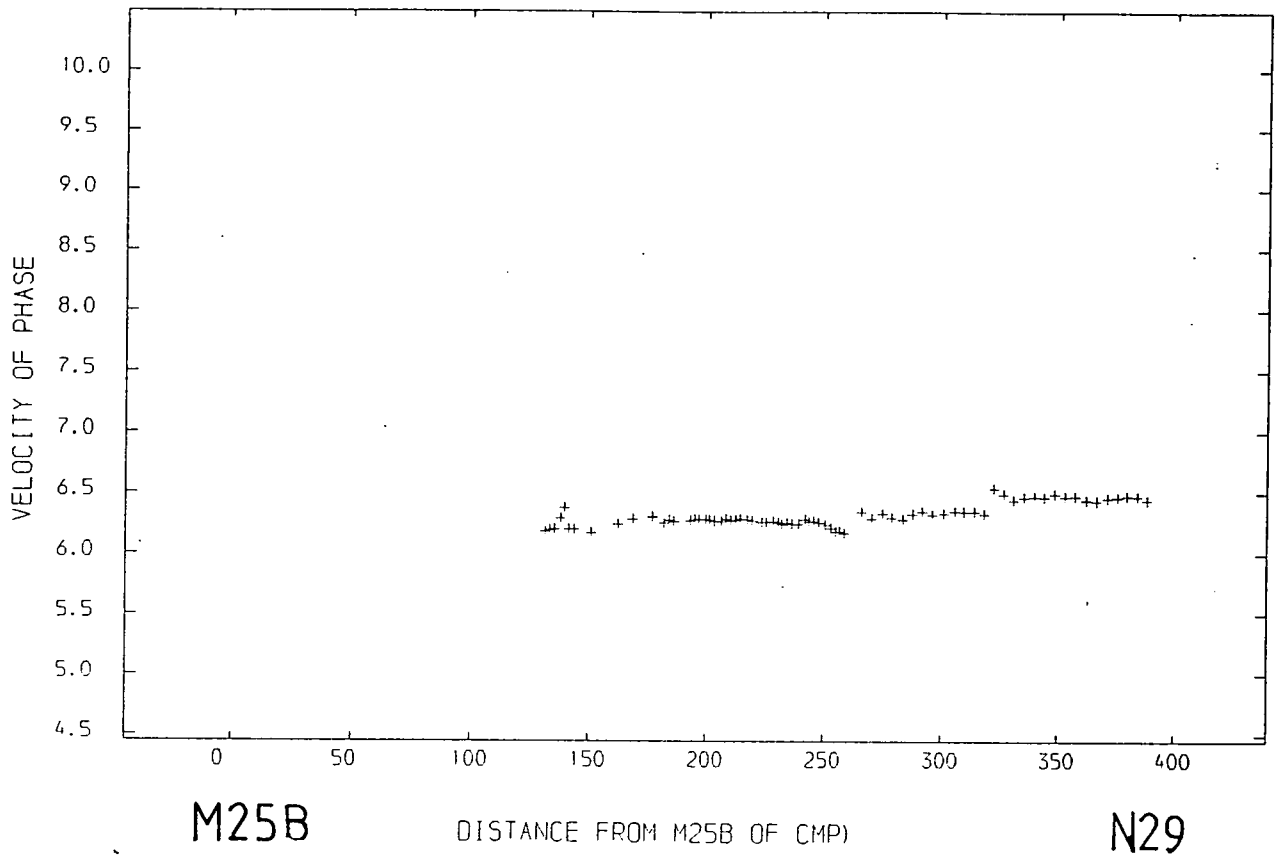


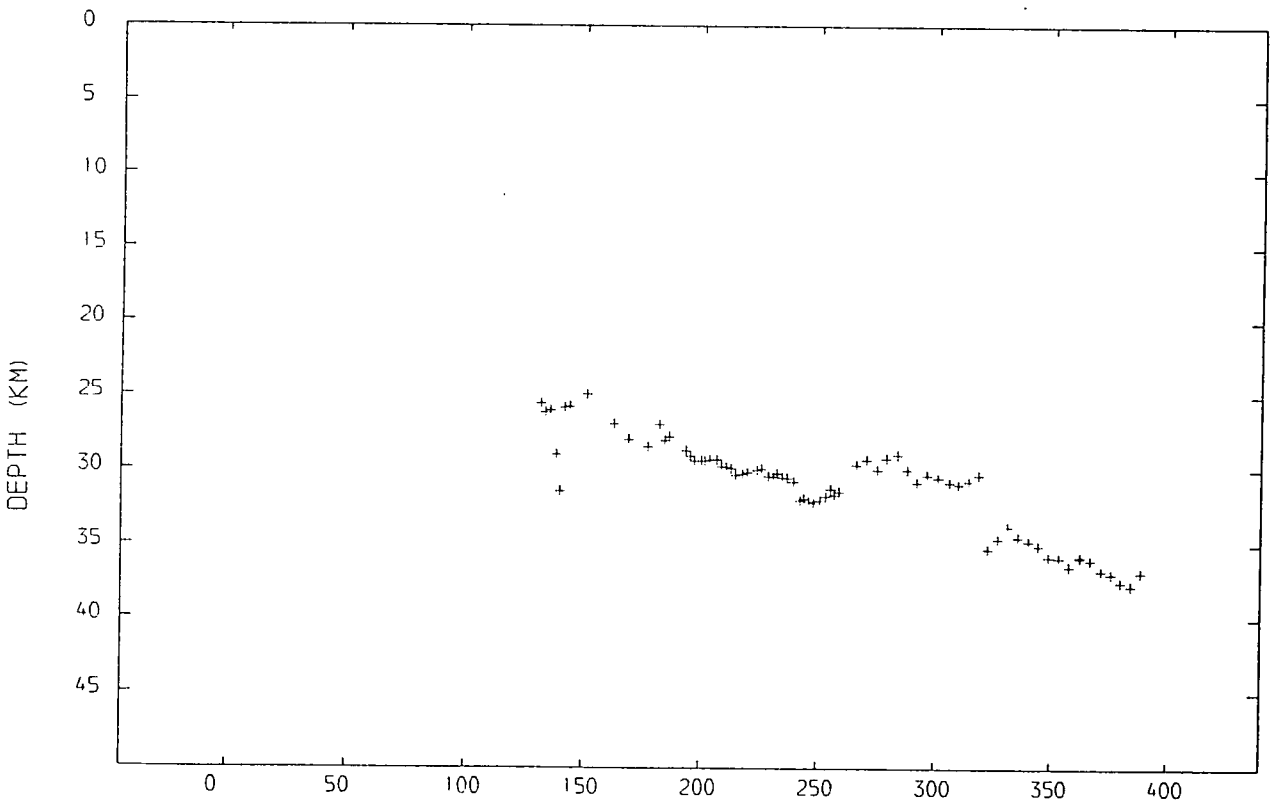
NSEA TT-REG RESULTS PCP



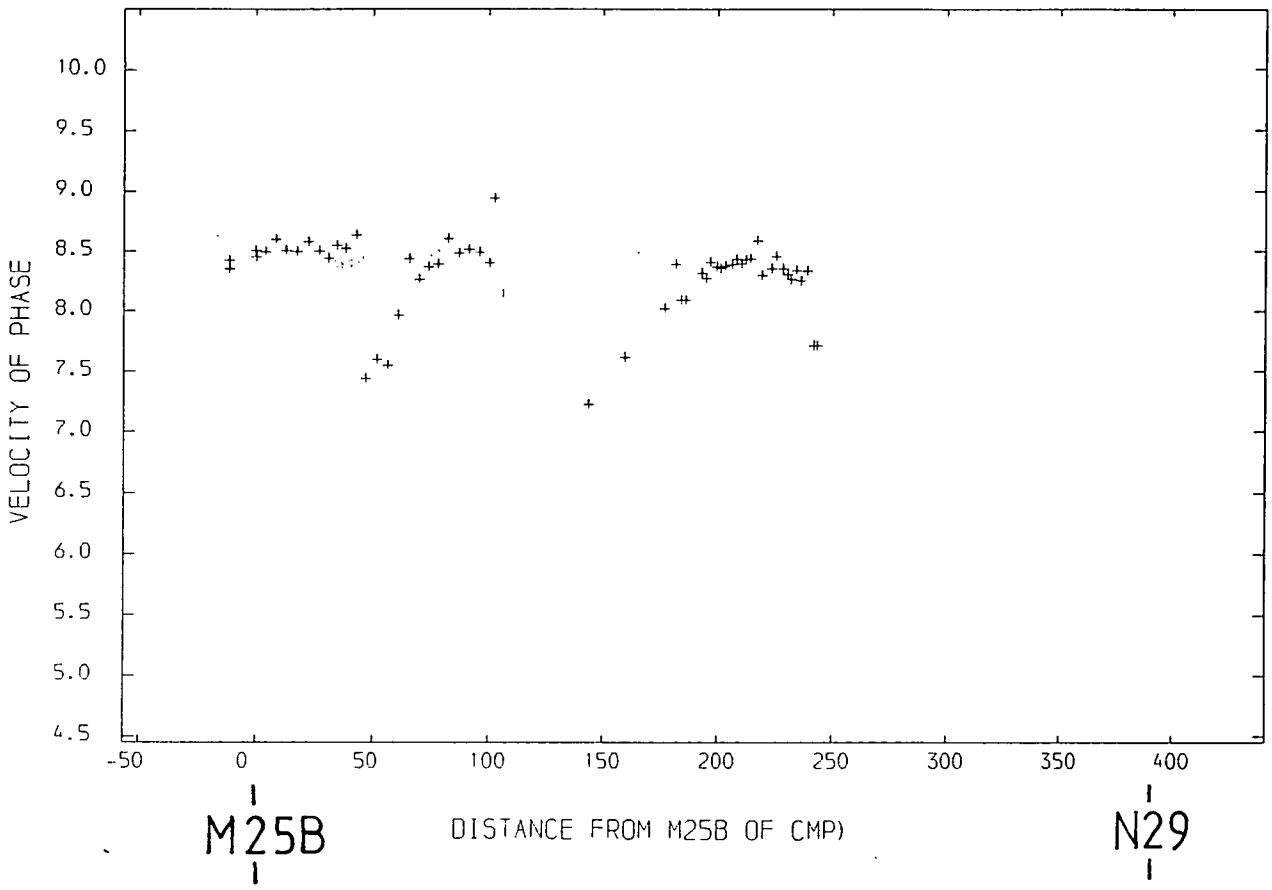


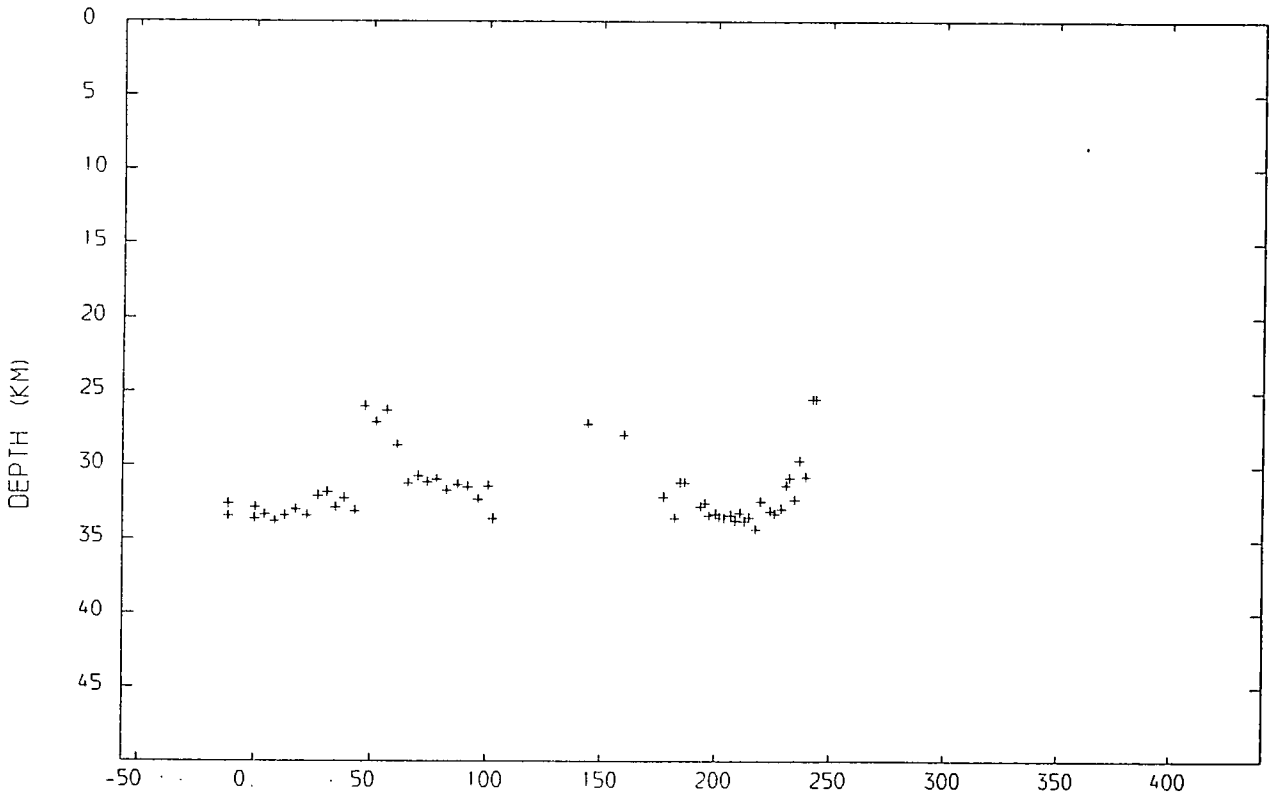
ISEA TT-REG RESULTS PMP



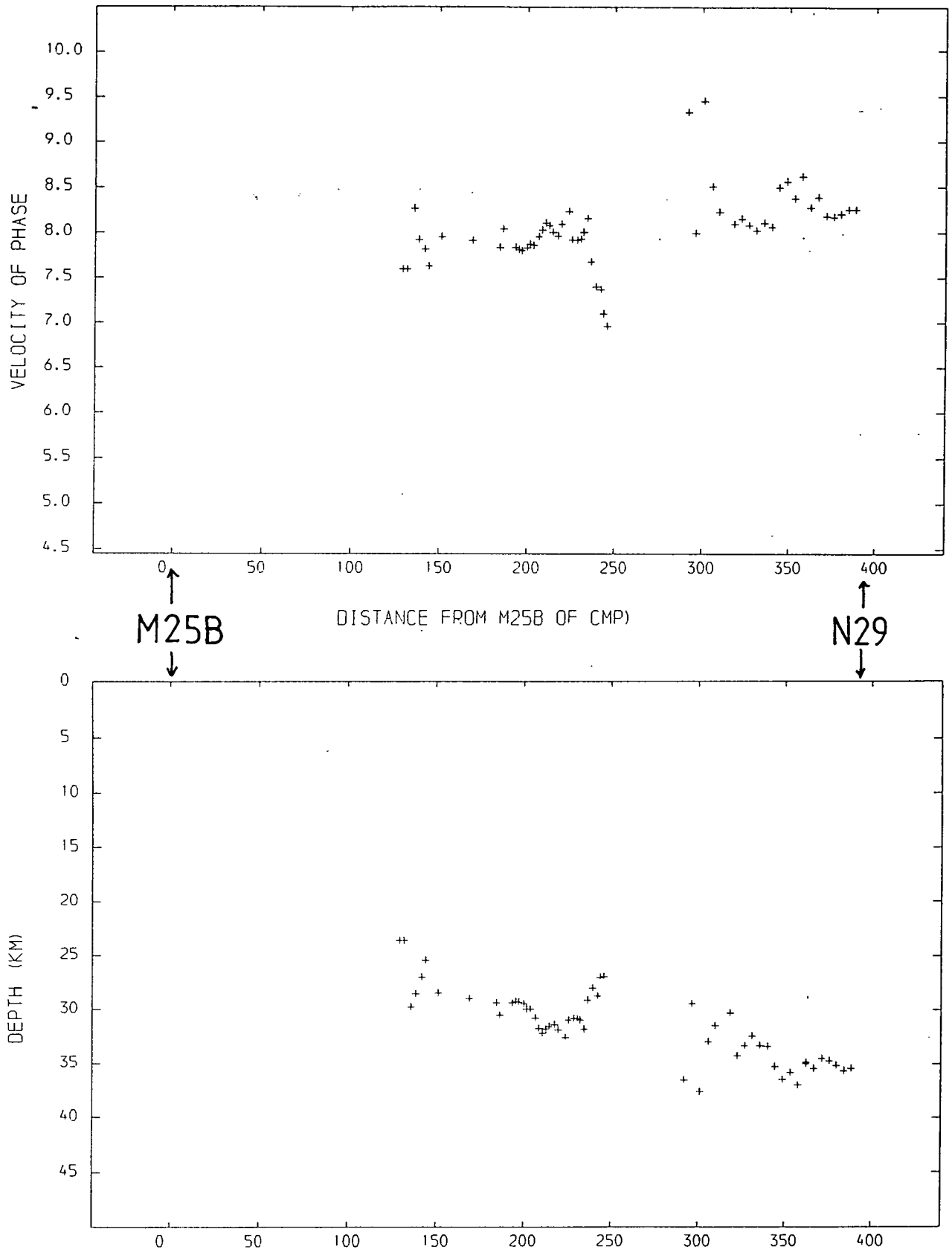


NSEA TT-REG RESULTS PMP

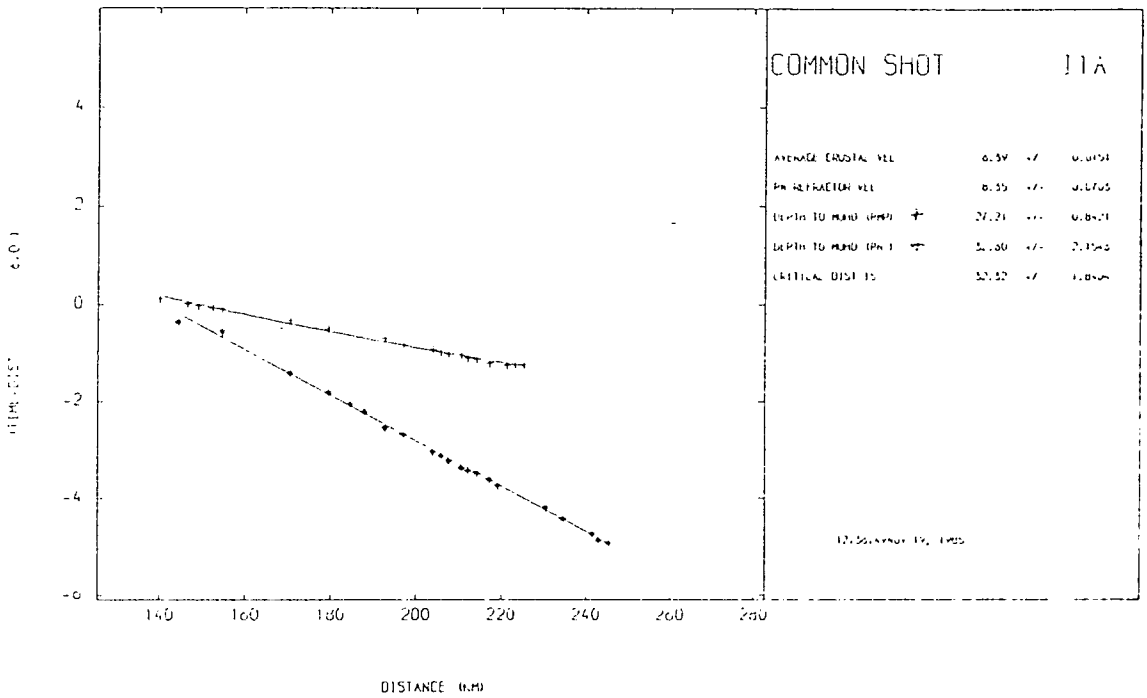


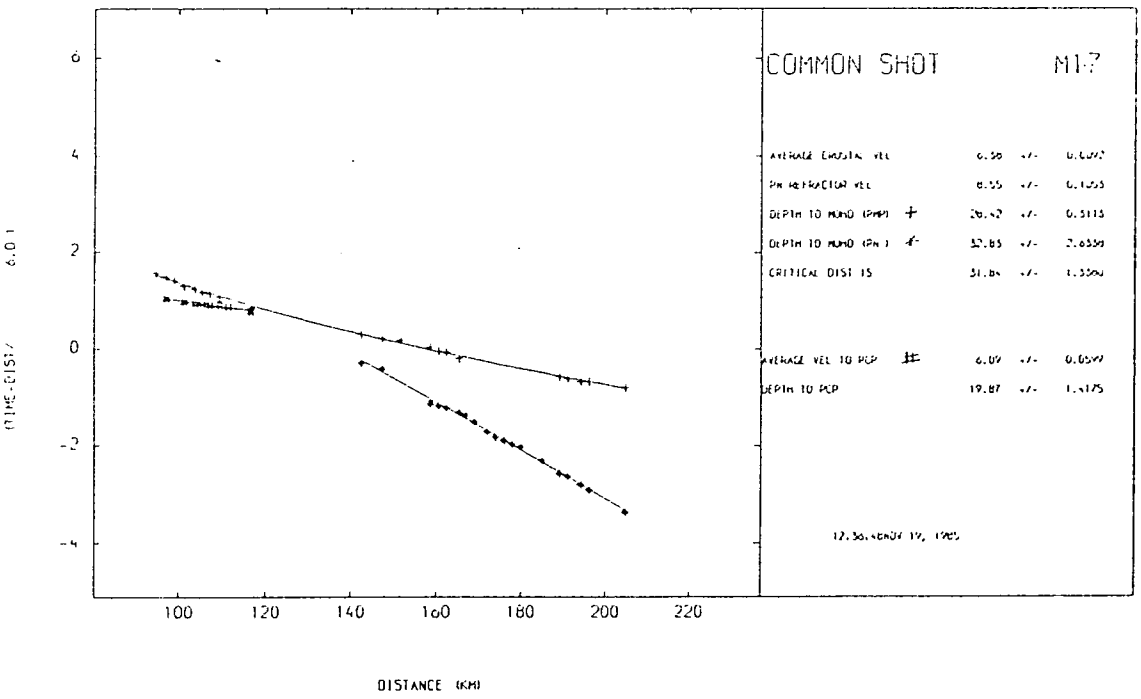


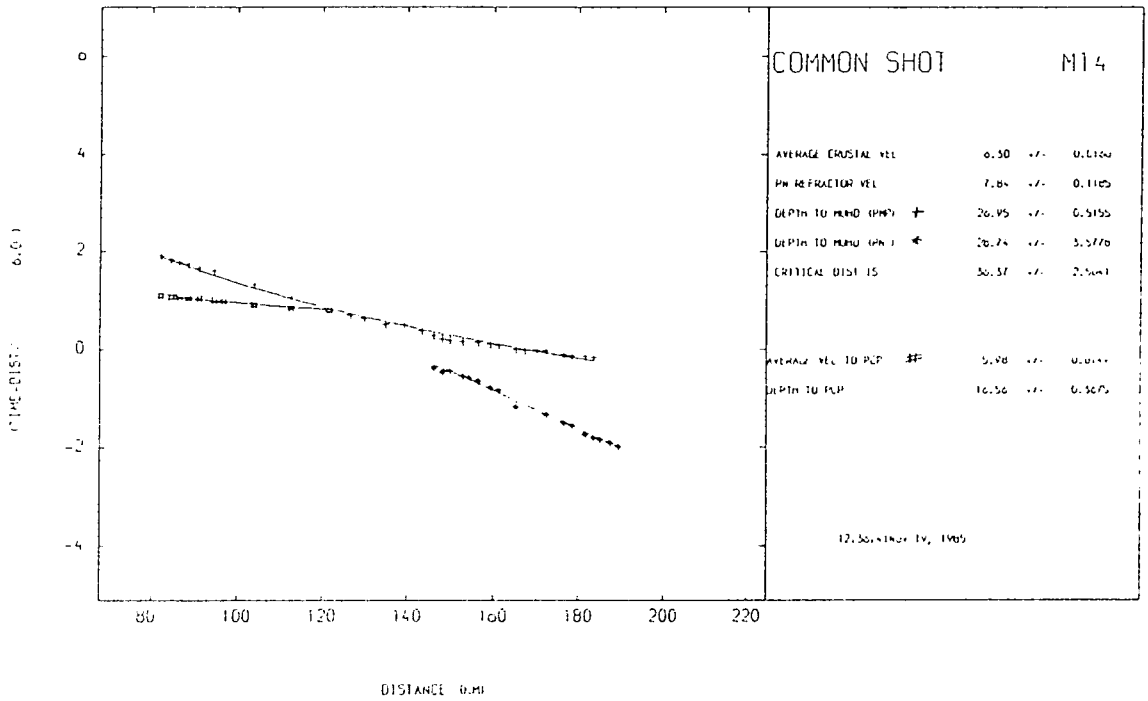
ISEA TT-REG RESULTS PN

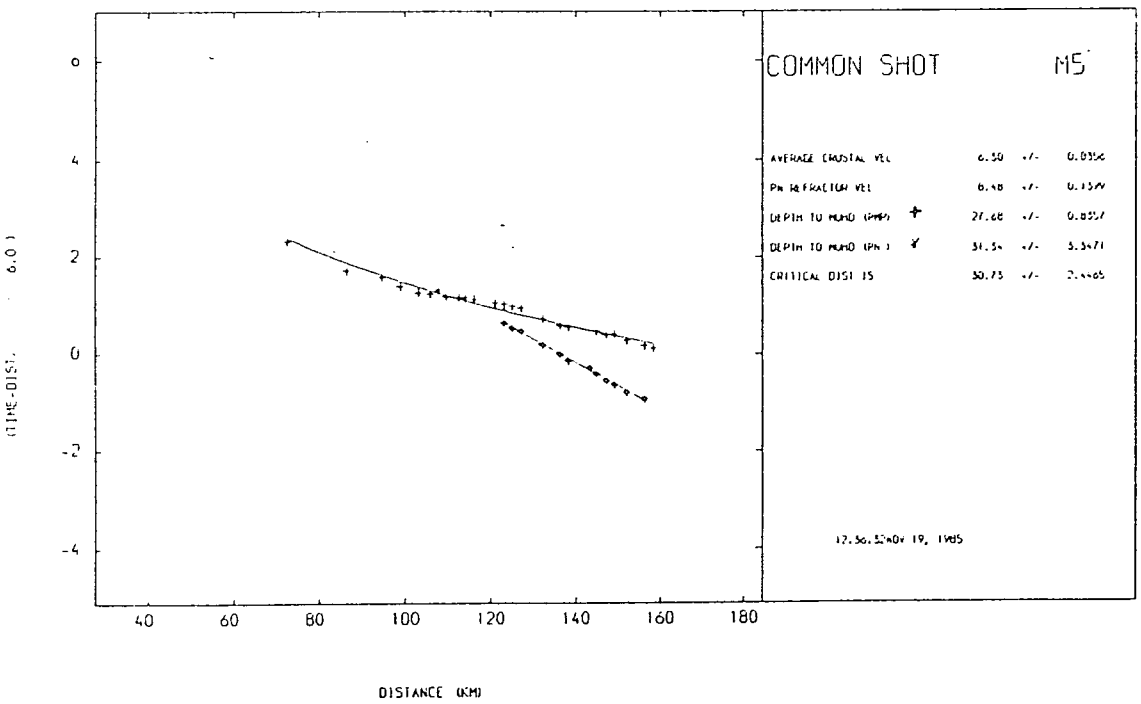


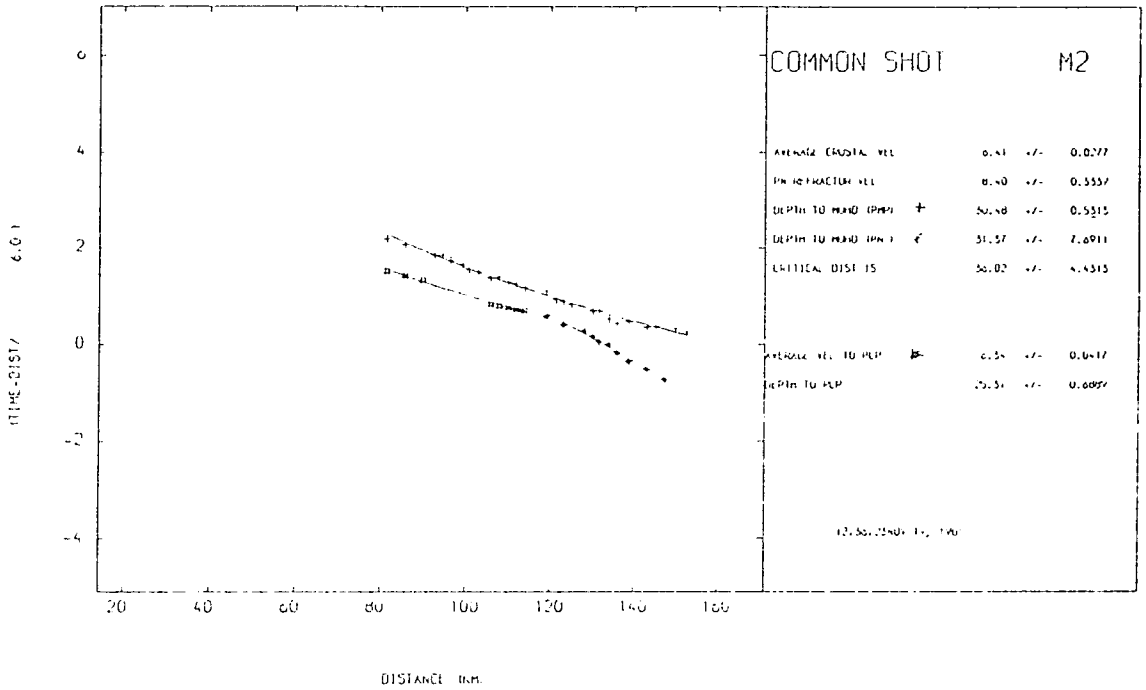
NSEA TT-REG RESULTS PN

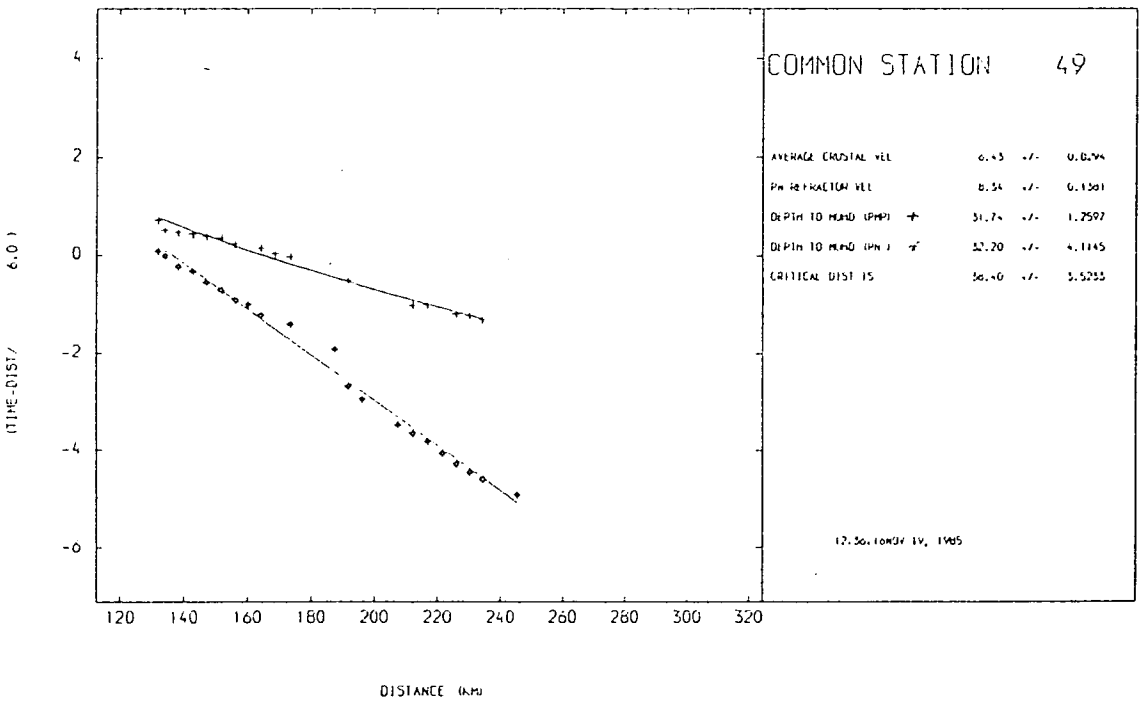

COMMON SHOT
11A
AVERAGE CRUSTAL VEL
PN REFRACTOR VEL
DEPTH TO MOHO (PMP)
DEPTH TO MOHO (PN)
CRITICAL DIST IS
TIME-DIST / 6.0
DISTANCE (KM)

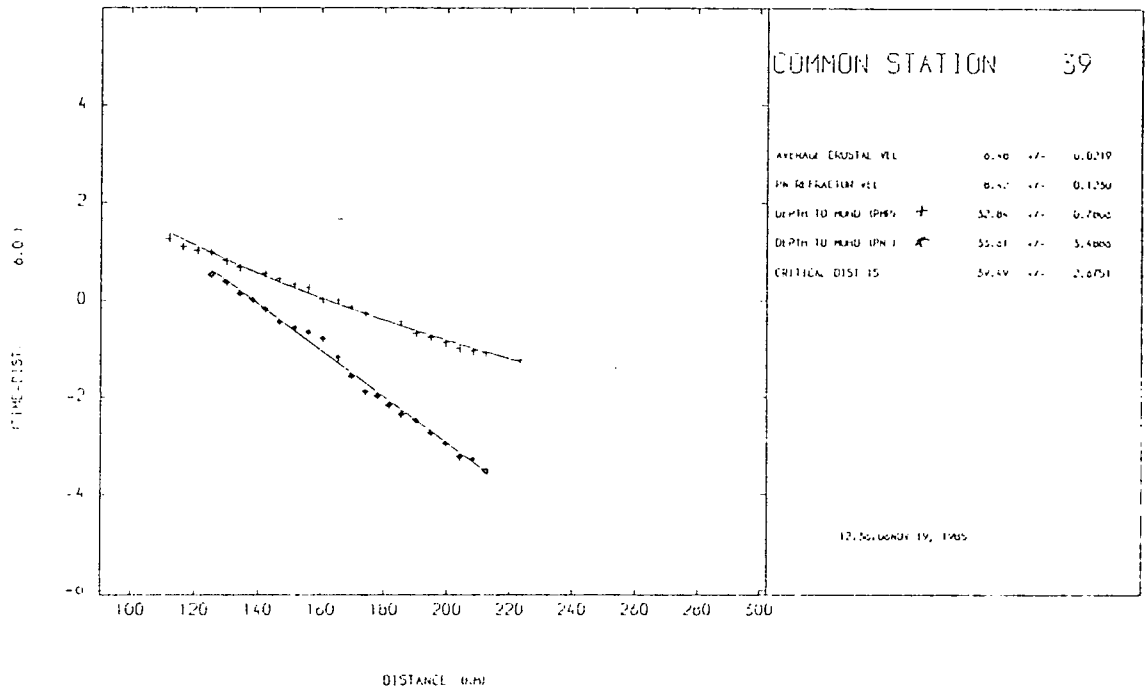

COMMON SHOT
M17
AVERAGE CRUSTAL VEL
PN REFRACTOR VEL
DEPTH TO MOHO (PMP)
DEPTH TO MOHO (PN)
CRITICAL DIST IS
AVERAGE VEL TO PCP
DEPTH TO PCP
TIME-DIST / 6.0
DISTANCE (KM)

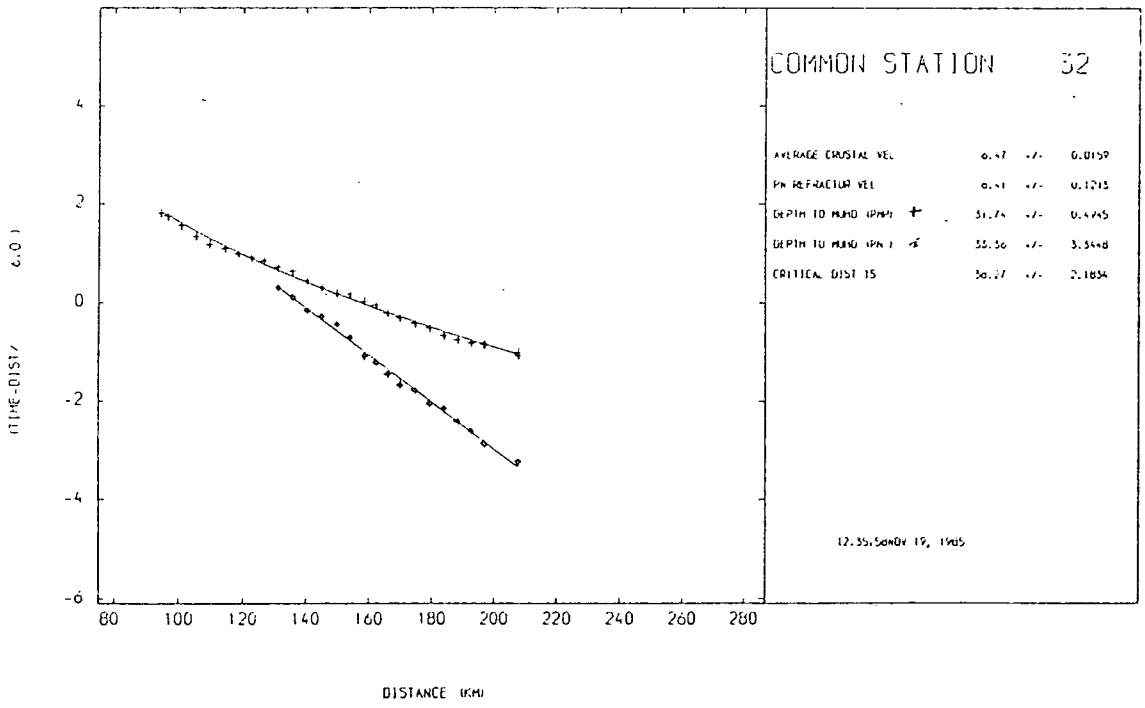

COMMON SHOT M14
AVERAGE CRUSTAL VEL 6.50 +/- 0.0126
Pn REFRACTOR VEL 7.84 +/- 0.1165
DEPTH TO MOHO (PMP) + 26.95 +/- 0.5155
DEPTH TO MOHO (Pn) 26.74 +/- 3.5776
CRITICAL DIST IS 50.37 +/-
AVERAGE VEL TO PCP # 5.98 +/-
DEPTH TO PCP 16.56 +/- 0.5675
12.56.41NOV 19, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)

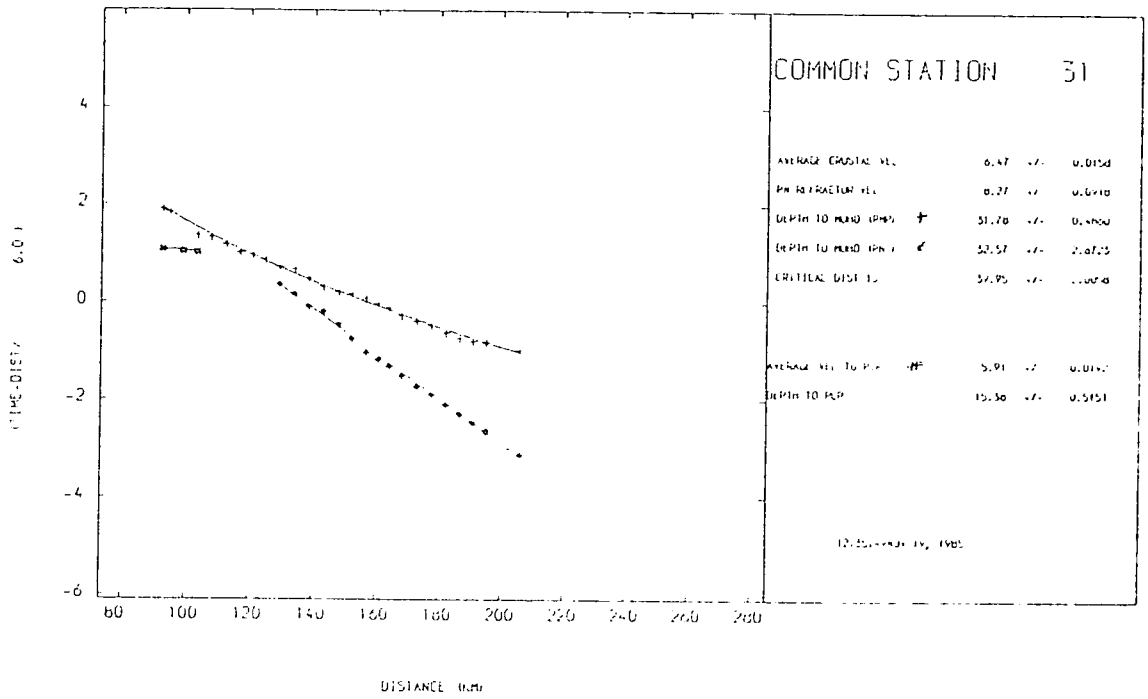

COMMON SHOT M5
AVERAGE CRUSTAL VEL 6.50 +/- 0.0356
Pn REFRACTOR VEL 8.48 +/- 0.1599
DEPTH TO MOHO (PMP) + 27.68 +/- 0.8557
DEPTH TO MOHO (Pn) 31.54 +/- 3.3471
CRITICAL DIST IS 50.73 +/-
12.56.52NOV 19, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)

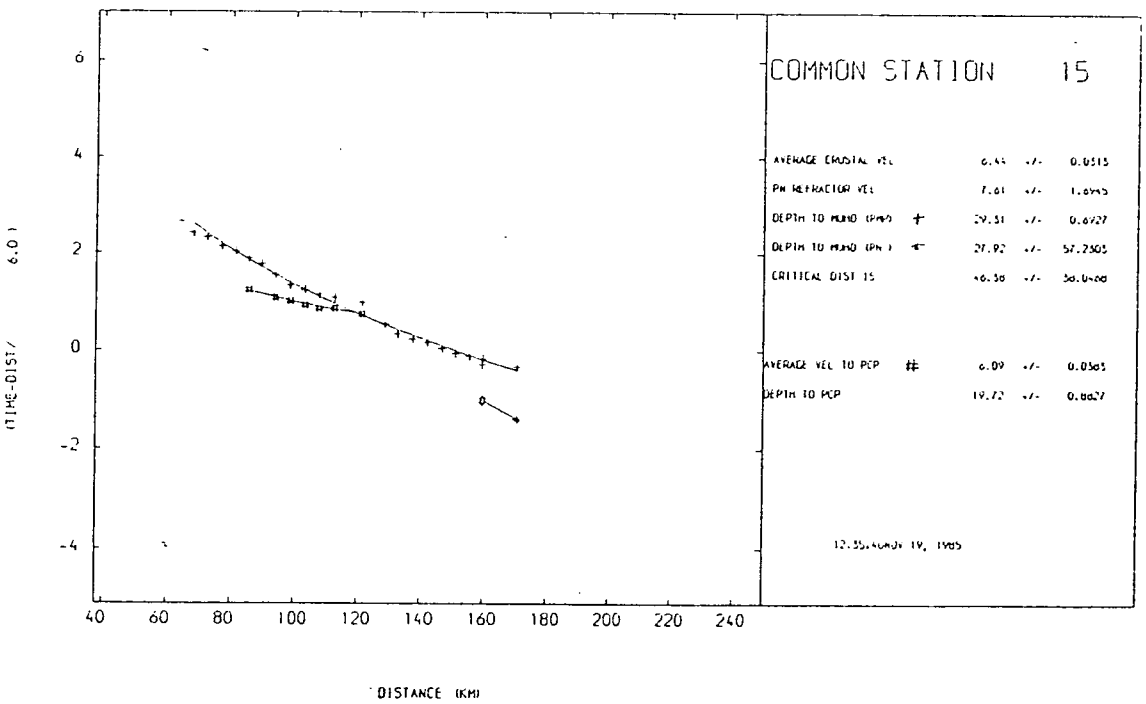

COMMON SHOT M2
AVERAGE CRUSTAL VEL 6.41 +/- 0.0277
Pn REFRACTION VEL 8.40 +/- 0.3557
DEPTH TO MOHO (PMP) + 50.48 +/- 0.5315
DEPTH TO MOHO (PN) 31.57 +/- 2.6911
CRITICAL DIST IS 56.02 +/- 4.4313
AVERAGE VEL TO PLP 6.54 +/- 0.0417
DEPTH TO PLP 25.54 +/- 0.6859
(TIME-DIST/ 6.0)
DISTANCE (KM)

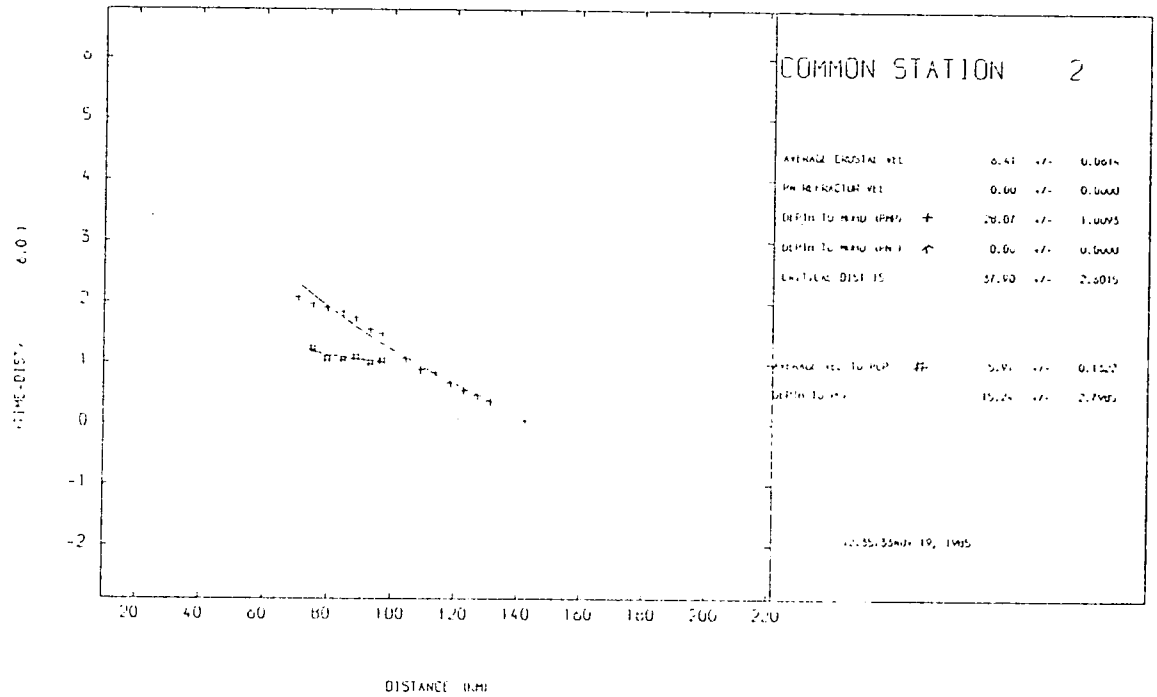

COMMON STATION 49
AVERAGE CRUSTAL VEL 6.43 +/- 0.0294
Pn REFRACTION VEL 8.54 +/- 0.1581
DEPTH TO MOHO (PMP) + 31.74 +/- 1.2597
DEPTH TO MOHO (PN) 32.20 +/- 4.1145
CRITICAL DIST IS 56.40 +/- 5.5255
(TIME-DIST/ 6.0)
DISTANCE (KM)

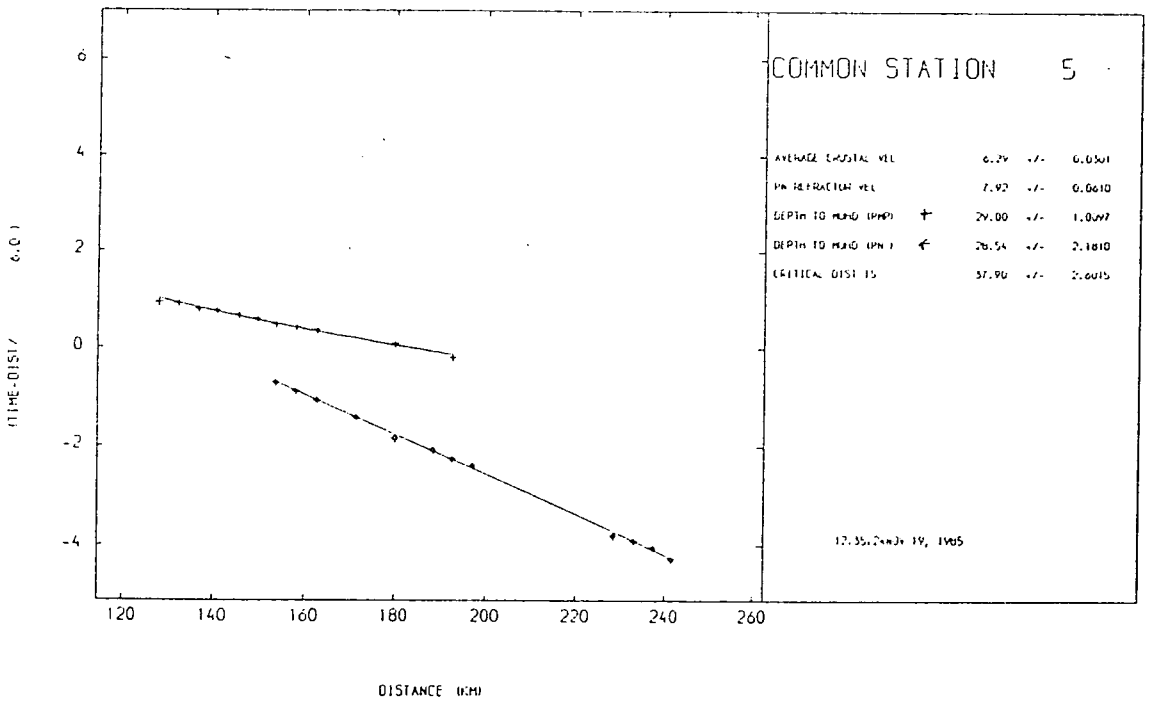

COMMON STATION 39
AVERAGE CRUSTAL VEL 0.48 +/- 0.0219
PN REFRACTOR VEL 0.82 +/- 0.1250
DEPTH TO MOHO (PMP) + 32.84 +/- 0.7602
DEPTH TO MOHO (PN) 55.51 +/- 3.4800
CRITICAL DIST IS 39.49 +/- 2.6751
12.36.06NOV 19, 1985
TIME-DIST (0.0)
DISTANCE (KM)

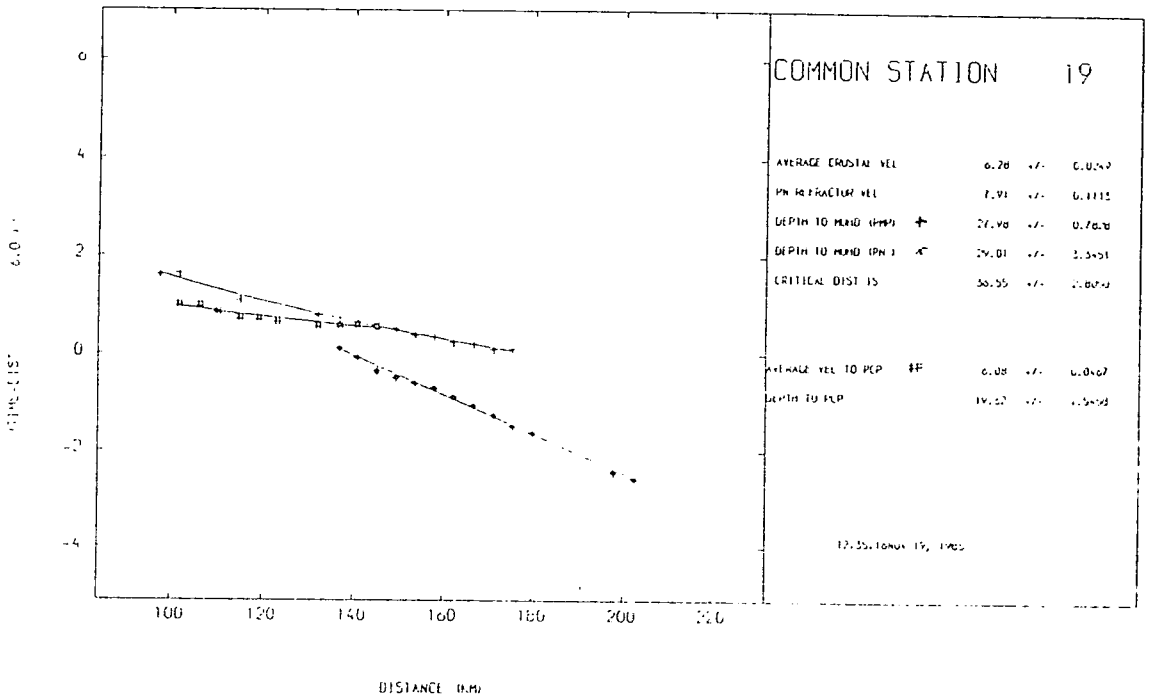

COMMON STATION 32
AVERAGE CRUSTAL VEL 0.47 +/- 0.0159
PN REFRACTOR VEL 0.81 +/- 0.1215
DEPTH TO MOHO (PMP) + 31.74 +/- 0.4745
DEPTH TO MOHO (PN) 55.56 +/- 3.3448
CRITICAL DIST IS 36.27 +/- 2.1834
12.35.50NOV 19, 1985
TIME-DIST (0.0)
DISTANCE (KM)

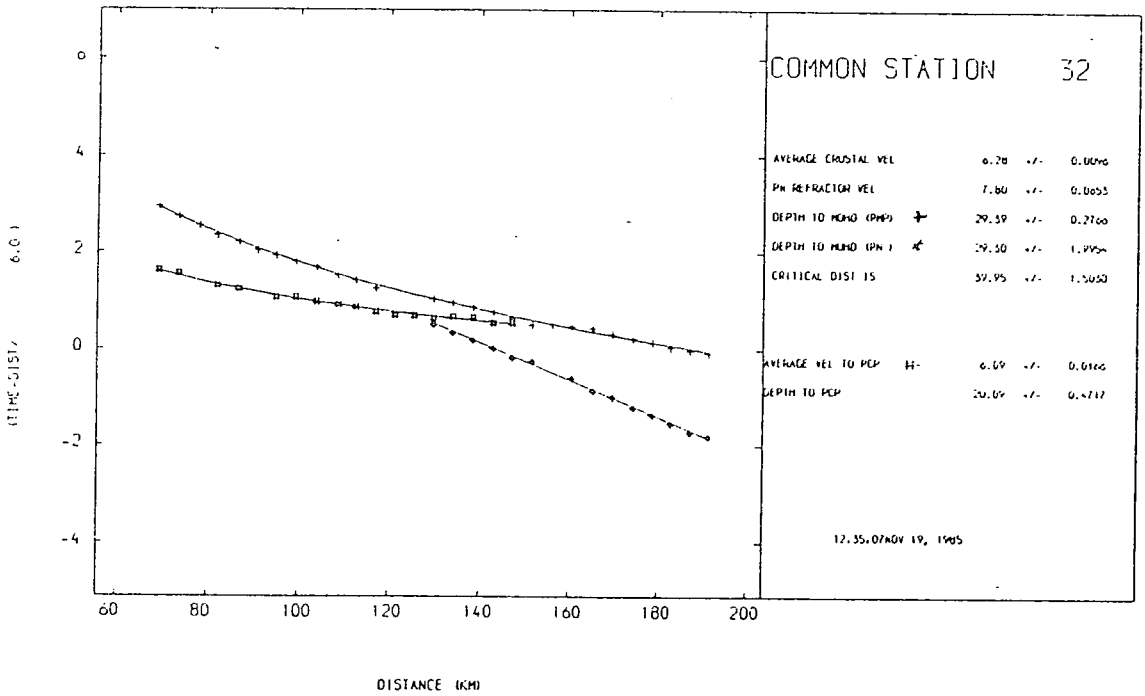

COMMON STATION 31
AVERAGE CRUSTAL VEL 6.47 +/- 0.0156
PM REFRACTOR VEL 8.27 +/- 0.0410
DEPTH TO MOHO (PMP) + 51.78 +/- 0.4900
DEPTH TO MOHO (PN) 52.57 +/- 2.6725
CRITICAL DIST 15 57.95 +/-
AVERAGE VEL TO PCP 5.91 +/-
DEPTH TO PCP 15.36 +/- 0.5151
(TIME-DIST/ 6.0)
DISTANCE (KM)

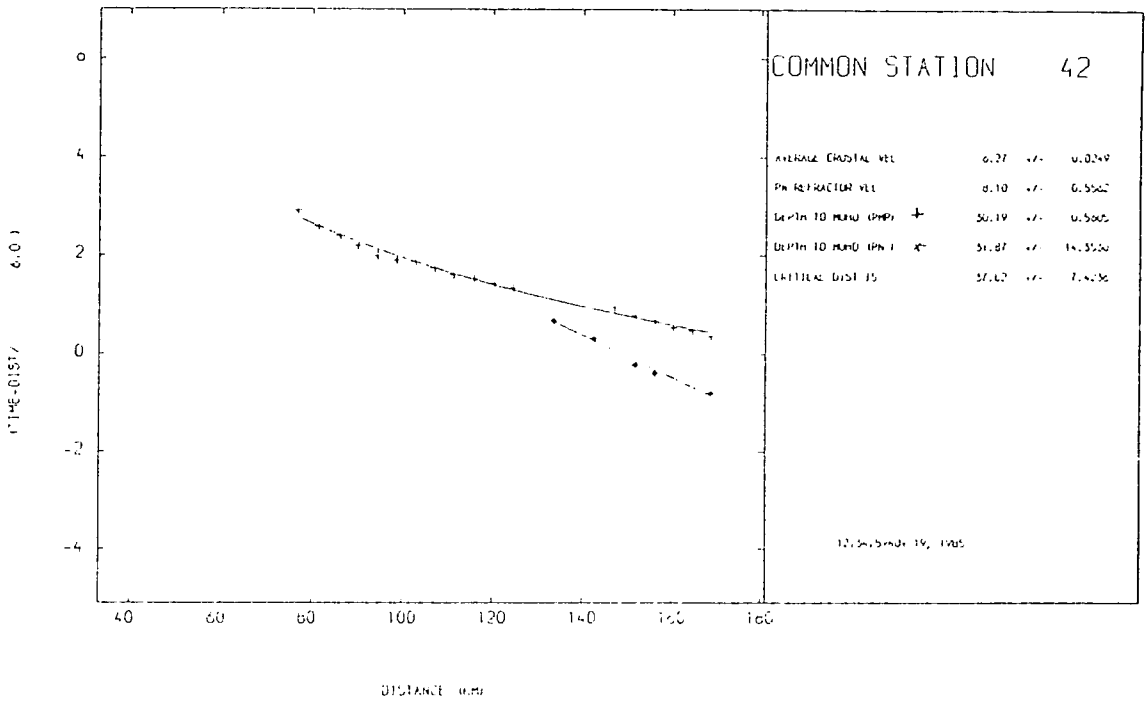

COMMON STATION 15
AVERAGE CRUSTAL VEL 6.45 +/- 0.0515
PM REFRACTOR VEL 7.61 +/- 1.0445
DEPTH TO MOHO (PMP) + 29.51 +/- 0.6727
DEPTH TO MOHO (PN) 27.92 +/- 57.2505
CRITICAL DIST 15 46.58 +/- 36.0400
AVERAGE VEL TO PCP 6.09 +/- 0.0365
DEPTH TO PCP 19.72 +/- 0.8027
(TIME-DIST/ 6.0)
DISTANCE (KM)

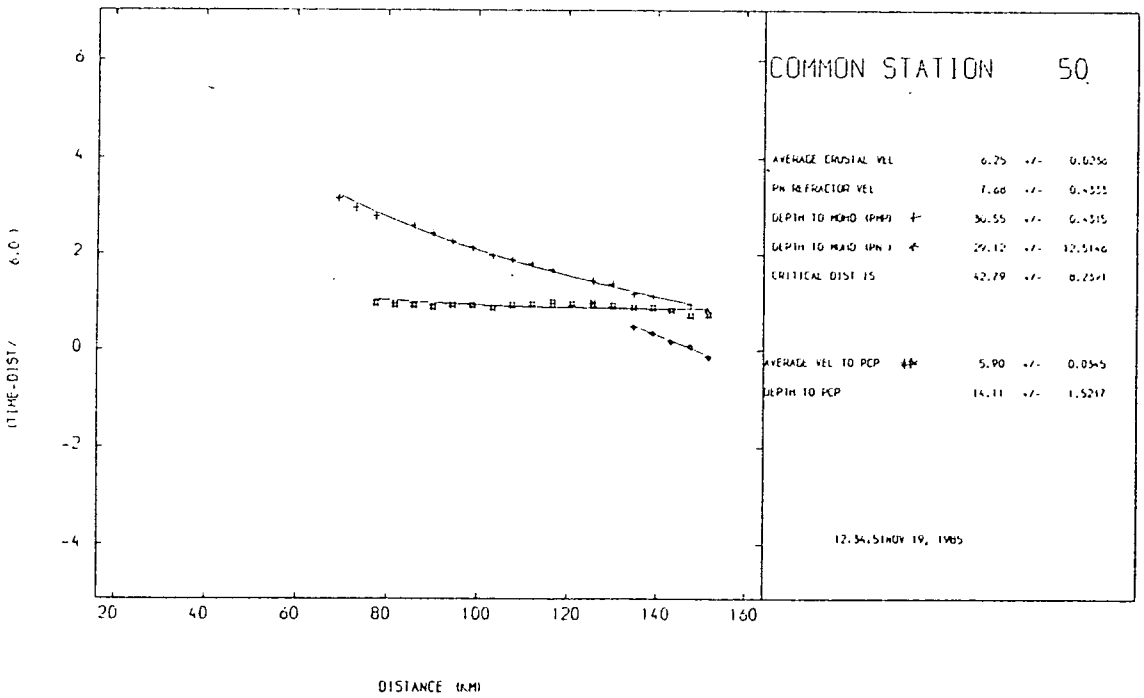

COMMON STATION 2
AVERAGE CRUSTAL VEL 6.41 +/- 0.0614
PM REFRACTOR VEL 0.00 +/- 0.0000
DEPTH TO MOHO (PMP) + 28.07 +/- 1.0095
DEPTH TO MOHO (PM) 0.00 +/- 0.0000
CRITICAL DIST IS 37.90 +/- 2.6015
DISTANCE (KM)

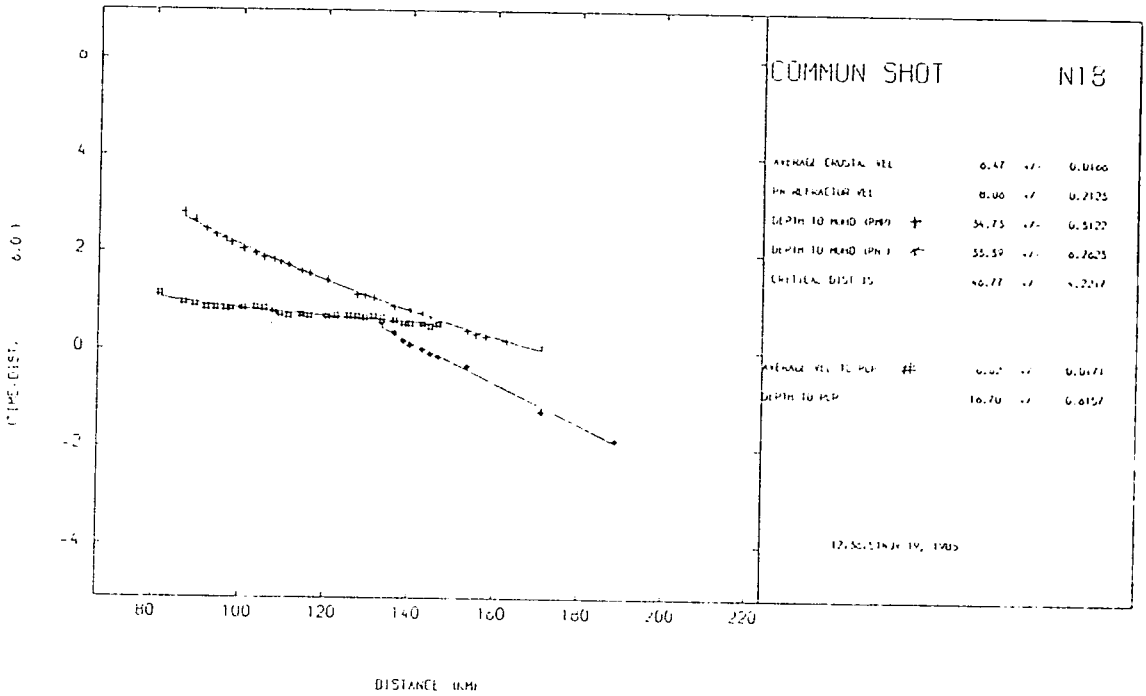

COMMON STATION 5
AVERAGE CRUSTAL VEL 6.29 +/- 0.0501
PM REFRACTOR VEL 7.92 +/- 0.0610
DEPTH TO MOHO (PMP) + 29.00 +/- 1.0097
DEPTH TO MOHO (PM) 26.54 +/- 2.1810
CRITICAL DIST IS 37.90 +/- 2.6015
DISTANCE (KM)

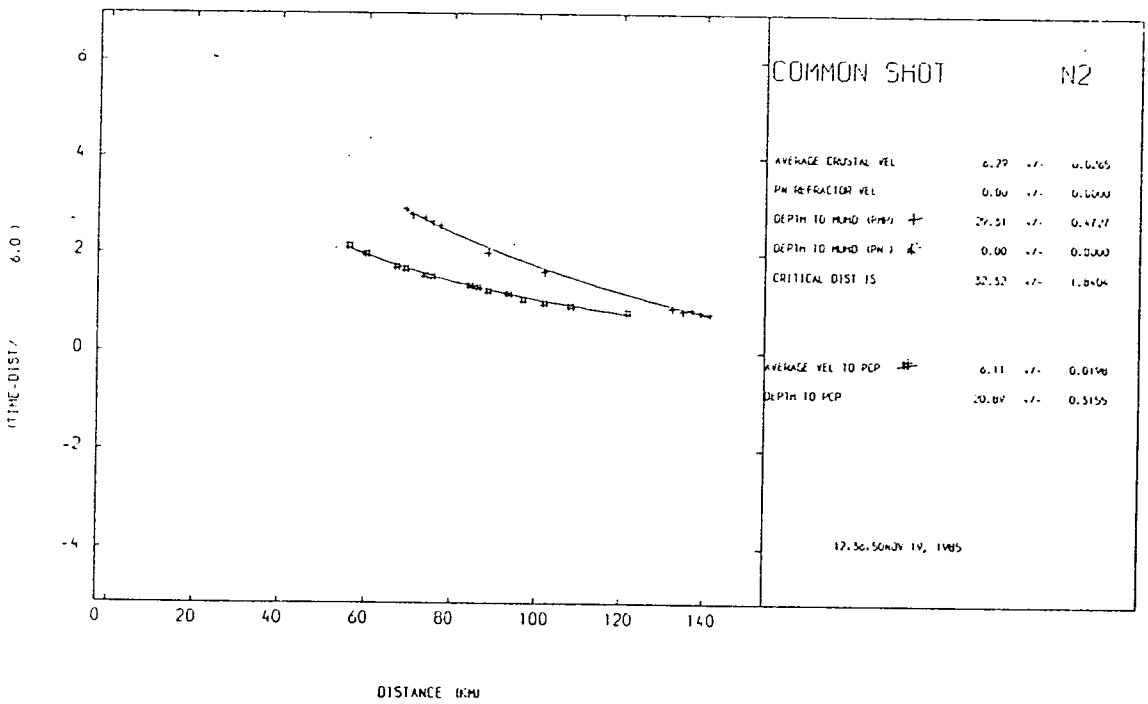

COMMON STATION 19
AVERAGE CRUSTAL VEL 6.28 +/- 0.0549
PN REFRACTOR VEL 7.91 +/- 0.1115
DEPTH TO MOHO (PMP) 27.98 +/- 0.7678
DEPTH TO MOHO (PN) 29.01 +/- 3.5451
CRITICAL DIST IS 38.55 +/- 2.8074
AVERAGE VEL TO PCP 6.08 +/- 0.0467
DEPTH TO PCP 19.22 +/- 1.5458
12.35.16NOV 19, 1985
DISTANCE (KM)

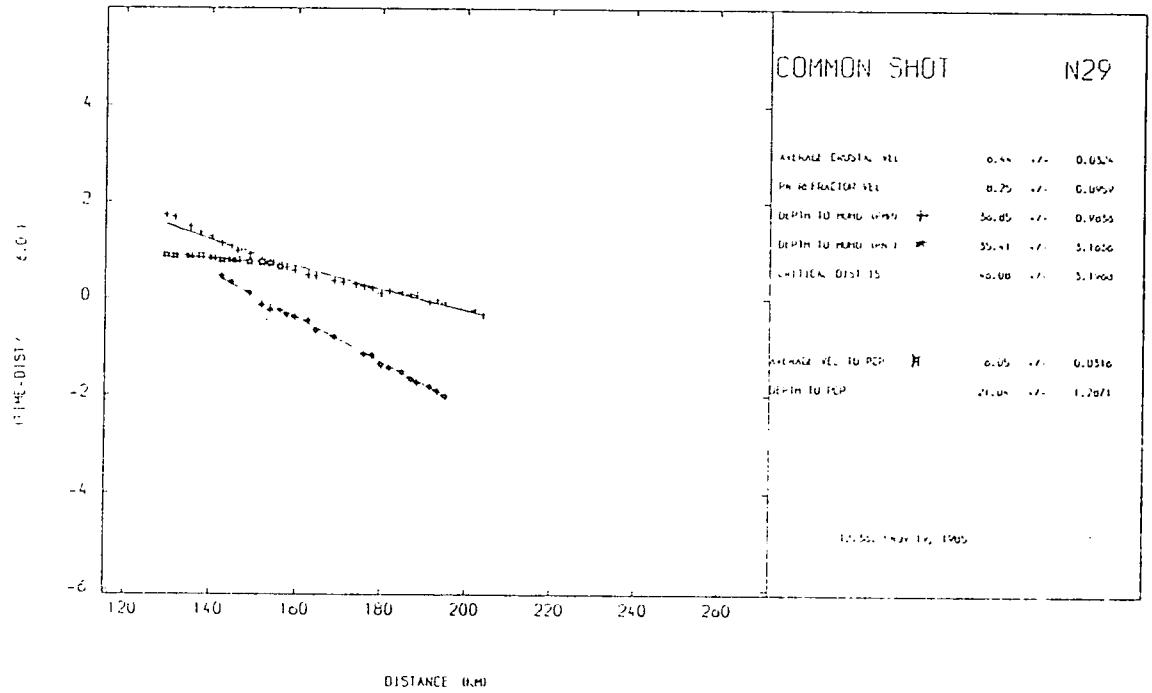

COMMON STATION 32
AVERAGE CRUSTAL VEL 6.28 +/- 0.0696
PN REFRACTOR VEL 7.60 +/- 0.0655
DEPTH TO MOHO (PMP) 29.39 +/- 0.2760
DEPTH TO MOHO (PN) 29.50 +/- 1.9954
CRITICAL DIST IS 37.95 +/- 1.5050
AVERAGE VEL TO PCP 6.09 +/- 0.0466
DEPTH TO PCP 20.09 +/- 0.4717
12.35.07NOV 19, 1985
DISTANCE (KM)

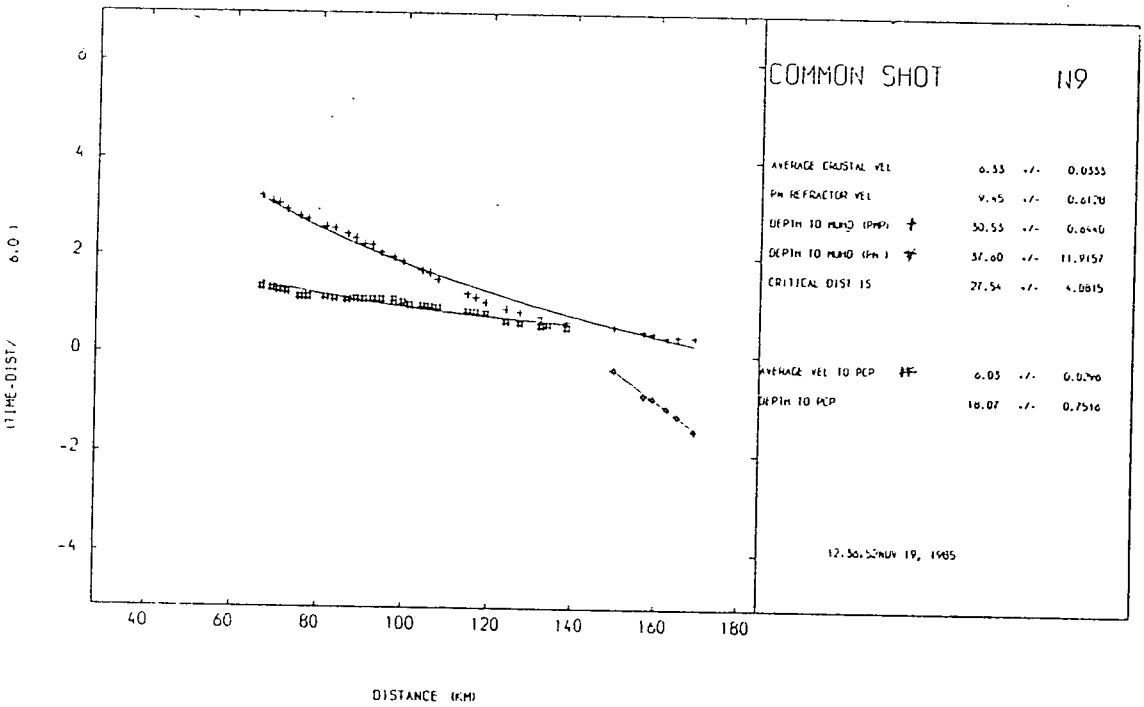

COMMON STATION 42
AVERAGE CRUSTAL VEL 6.27 +/- 0.0249
PN REFRACTOR VEL 8.10 +/- 0.5562
DEPTH TO MOHO (PMP) + 30.19 +/- 0.5605
DEPTH TO MOHO (PN) * 31.87 +/- 14.3550
CRITICAL DIST IS 37.62 +/- 7.4236
12:34:5 NOV 19, 1985
TIME-DIST/ 6.0)
DISTANCE (KM)

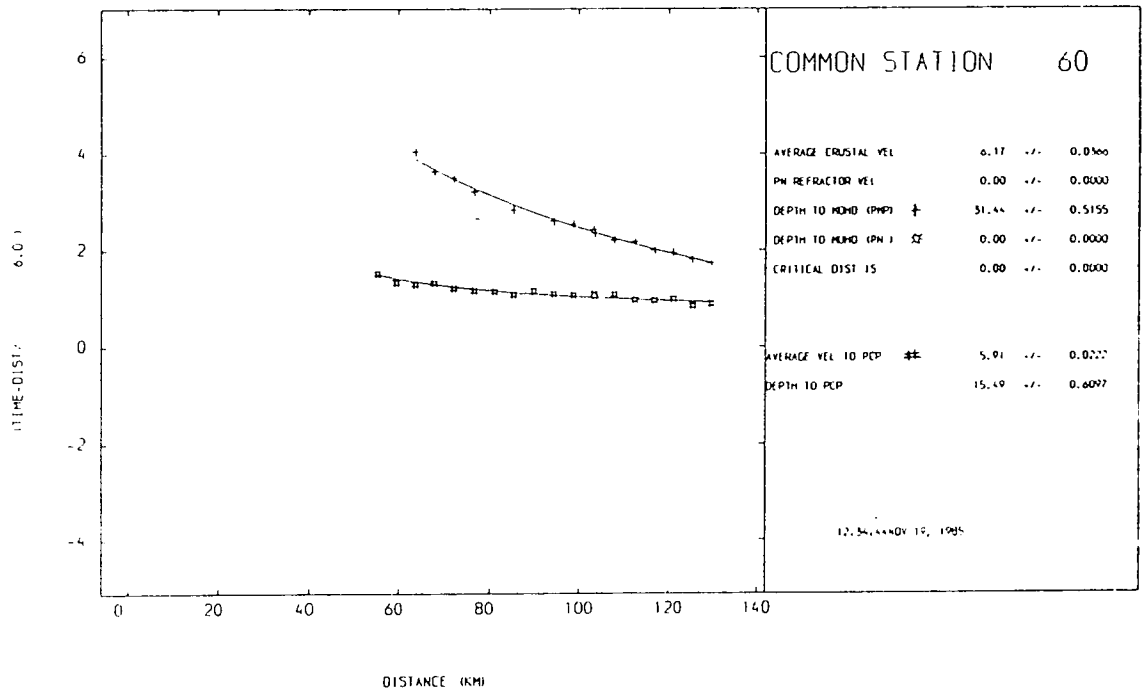

COMMON STATION 50
AVERAGE CRUSTAL VEL 6.25 +/- 0.0256
PN REFRACTOR VEL 7.68 +/- 0.4555
DEPTH TO MOHO (PMP) + 30.55 +/- 0.4515
DEPTH TO MOHO (PN) * 29.12 +/- 12.5146
CRITICAL DIST IS 42.79 +/- 8.2541
AVERAGE VEL TO PCP ** 5.90 +/- 0.0345
DEPTH TO PCP 14.11 +/- 1.5217
12:34:5 NOV 19, 1985
TIME-DIST/ 6.0)
DISTANCE (KM)

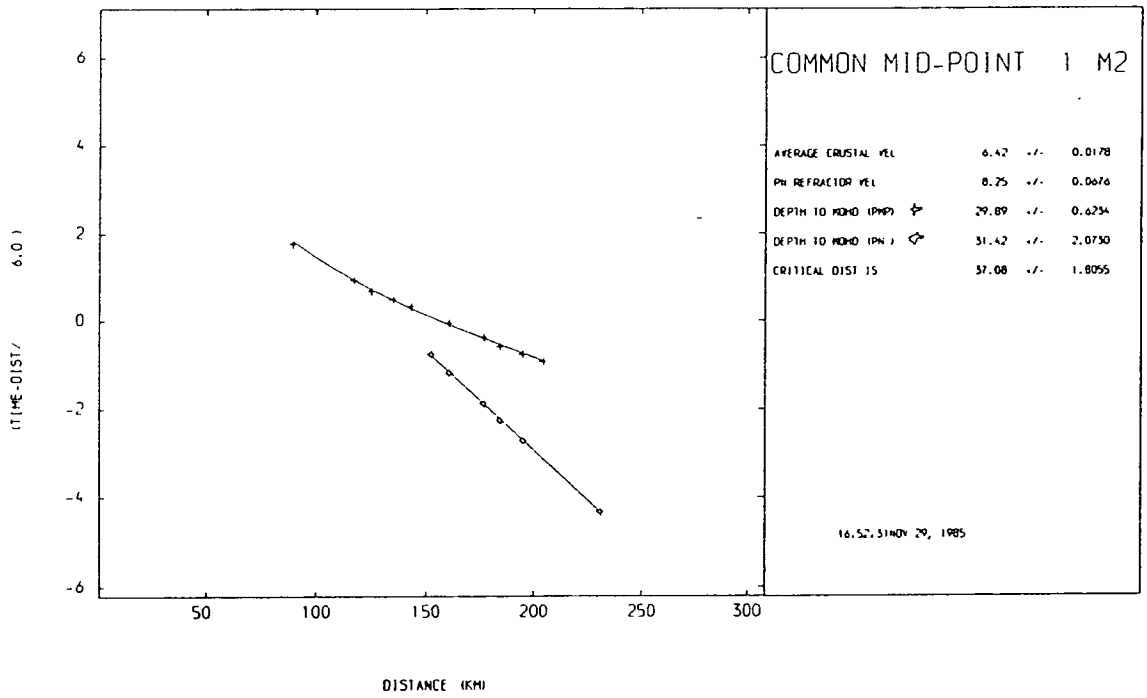

COMMON SHOT N18
AVERAGE CRUSTAL VEL 6.47 +/- 0.0166
PN REFRACTOR VEL 8.06 +/- 0.2135
DEPTH TO MOHO (PMP) + 34.73 +/- 0.5122
DEPTH TO MOHO (PN) 55.59 +/- 0.7625
CRITICAL DIST IS 46.77 +/- 1.2247
AVERAGE VEL TO PCP # 6.02 +/- 0.0171
DEPTH TO PCP 16.70 +/- 0.0157
12:36:11NOV 19, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)

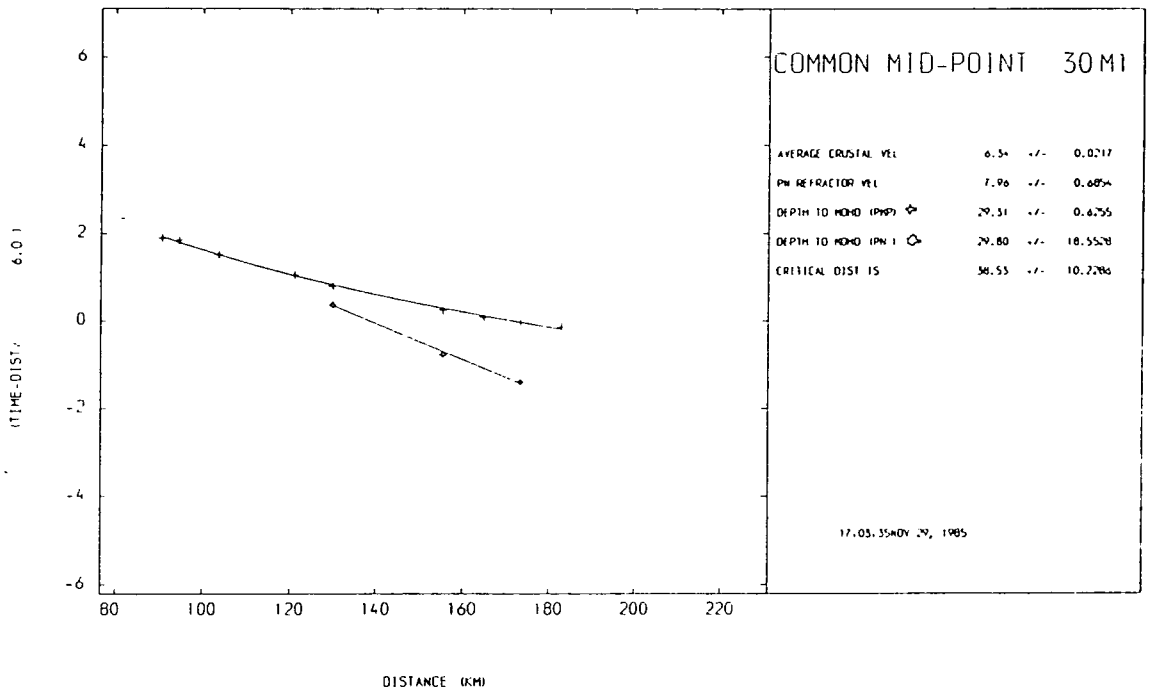

COMMON SHOT N2
AVERAGE CRUSTAL VEL 6.29 +/- 0.0265
PN REFRACTOR VEL 0.00 +/- 0.0000
DEPTH TO MOHO (PMP) + 29.51 +/- 0.4727
DEPTH TO MOHO (PN) 0.00 +/- 0.0000
CRITICAL DIST IS 32.52 +/- 1.8404
AVERAGE VEL TO PCP # 6.11 +/- 0.0198
DEPTH TO PCP 20.89 +/- 0.5155
12:36:50NOV 19, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)

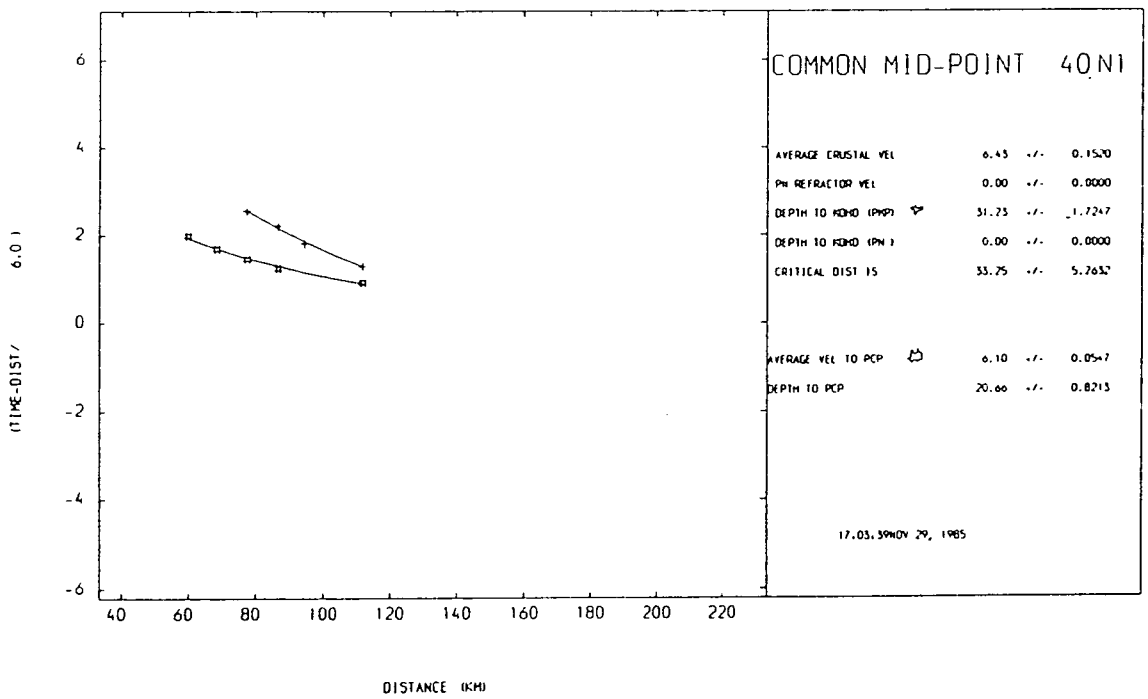

COMMON SHOT
N29
AVERAGE CRUSTAL VEL
PN REFRACTOR VEL
DEPTH TO MOHO (PMP)
DEPTH TO MOHO (PN)
CRITICAL DIST IS
AVERAGE VEL TO PCP
DEPTH TO PCP
+/-
TIME-DIST/ 6.0
DISTANCE (KM)

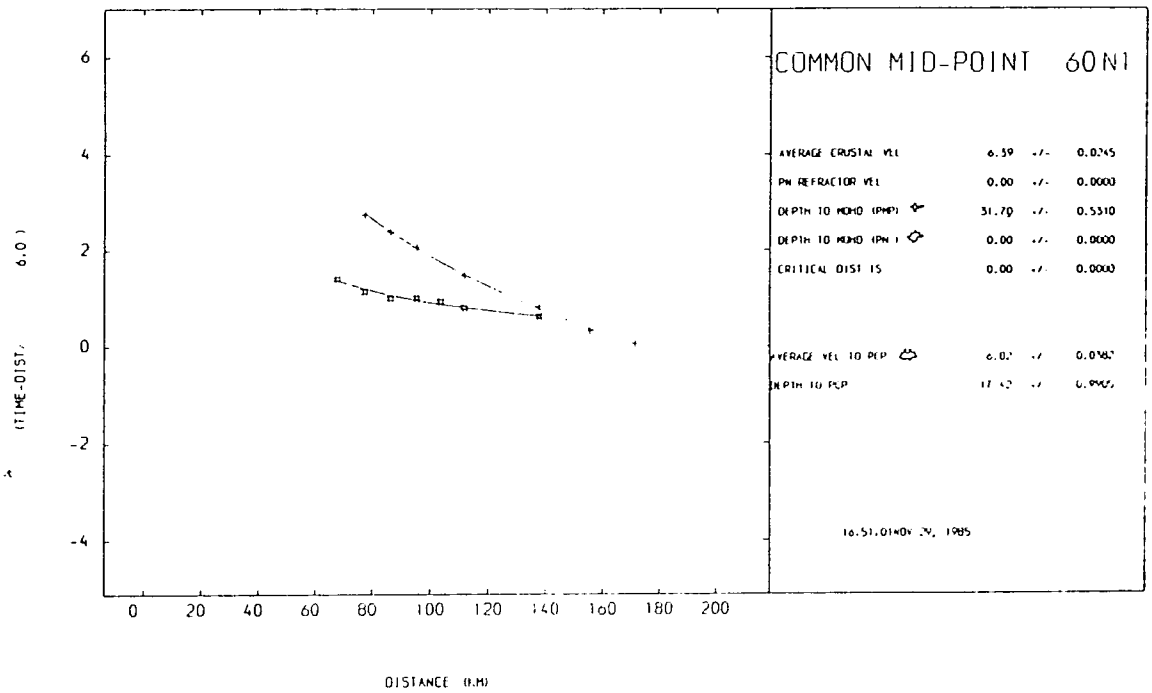

COMMON SHOT
N9
AVERAGE CRUSTAL VEL 6.33 +/- 0.0333
PN REFRACTOR VEL 7.45 +/- 0.0120
DEPTH TO MOHO (PMP) 50.53 +/- 0.0440
DEPTH TO MOHO (PN) 57.60 +/- 11.9157
CRITICAL DIST IS 27.54 +/- 4.0815
AVERAGE VEL TO PCP 6.03 +/- 0.0240
DEPTH TO PCP 18.07 +/- 0.7516
12.36.52NOV 19, 1985
TIME-DIST/ 6.0
DISTANCE (KM)

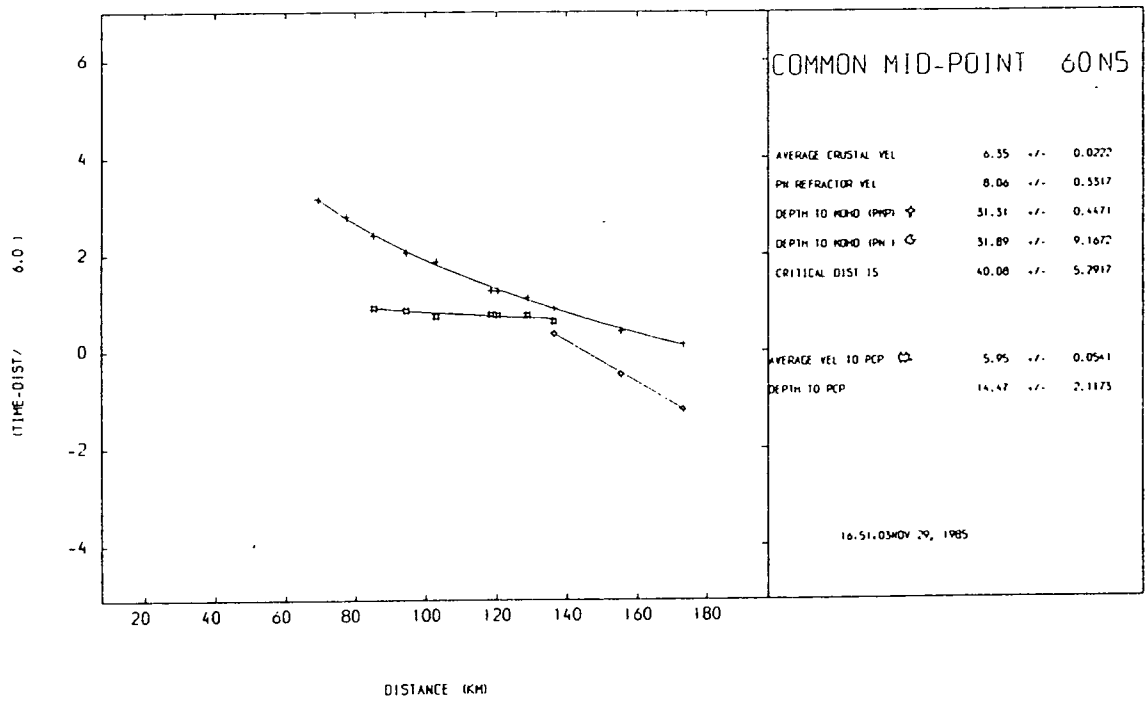

COMMON STATION 60
AVERAGE CRUSTAL VEL 6.17 +/- 0.0366
PN REFRACTOR VEL 0.00 +/- 0.0000
DEPTH TO MOHO (PMP) 31.44 +/- 0.5155
DEPTH TO MOHO (PN) 0.00 +/- 0.0000
CRITICAL DIST IS 0.00 +/- 0.0000
AVERAGE VEL TO PCP 5.91 +/- 0.0222
DEPTH TO PCP 15.49 +/- 0.6097
12.34.44NOV 19, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)


COMMON MID-POINT 1 M2
AVERAGE CRUSTAL VEL 6.42 +/- 0.0178
PN REFRACTOR VEL 8.25 +/- 0.0676
DEPTH TO MOHO (PMP) 29.89 +/- 0.6234
DEPTH TO MOHO (PN) 31.42 +/- 2.0730
CRITICAL DIST IS 37.08 +/- 1.8055
16.52.31NOV 29, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)


COMMON MID-POINT 30 M1
AVERAGE CRUSTAL VEL 6.54 +/- 0.0217
PN REFRACTOR VEL 7.96 +/- 0.6854
DEPTH TO MOHO (PMP) 29.51 +/- 0.6755
DEPTH TO MOHO (PN) 29.80 +/- 18.5528
CRITICAL DIST IS 58.53 +/- 10.2286
17.03.35NOV 29, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)


COMMON MID-POINT 40 N1
AVERAGE CRUSTAL VEL 6.45 +/- 0.1520
PN REFRACTOR VEL 0.00 +/- 0.0000
DEPTH TO MOHO (PMP) 31.23 +/- 1.7247
DEPTH TO MOHO (PN) 0.00 +/- 0.0000
CRITICAL DIST IS 53.75 +/- 5.7652
AVERAGE VEL TO PCP 6.10 +/- 0.0547
DEPTH TO PCP 20.66 +/- 0.8213
17.03.39NOV 29, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)


COMMON MID-POINT 60N1
AVERAGE CRUSTAL VEL 6.39 +/- 0.0245
PN REFRACTOR VEL 0.00 +/- 0.0000
DEPTH TO MOHO (PMP) 31.70 +/- 0.5310
DEPTH TO MOHO (PN) 0.00 +/- 0.0000
CRITICAL DIST IS 0.00 +/- 0.0000
AVERAGE VEL TO PCP
DEPTH TO PCP
16.51.01NOV 29, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)


COMMON MID-POINT 60N5
AVERAGE CRUSTAL VEL 6.35 +/- 0.0222
PN REFRACTOR VEL 8.06 +/- 0.5317
DEPTH TO MOHO (PMP) 31.31 +/- 0.4471
DEPTH TO MOHO (PN) 31.89 +/- 9.1672
CRITICAL DIST IS 40.08 +/- 5.2917
AVERAGE VEL TO PCP 5.95 +/- 0.0541
DEPTH TO PCP 14.47 +/- 2.1173
16.51.03NOV 29, 1985
(TIME-DIST/ 6.0)
DISTANCE (KM)

Appendix D

PROGRAMME LISTINGS

All programmes are held in the file GPT9 CSSP PROG. Example inputs mentioned in the documentation to each programme can be found in the file GPT9 CSSP DAT. The following programmes are included in this appendix.

PROGRAMME	DESCRIPTION	PAGE
TR-INPUT.U	- creates an input and source file for TR-CSP	
TR-CSP.U	- reads DSDIF tapes and copies traces to disc file	
TR-IGS.U	- reads Edinburgh tapes and copies traces to disc file	
CDIST.U	- corrects distances for a CDS section	
CSSPLOT.U	- plots traces read from tape by TR-CSP and TR-IGS	
TMATCOPY.U	- copies travel times to the matrix files	
TMATREAD.U	- reads travel times in the matrix files	
TMATRD-CST.U	- reads CST travel times onto disc for use with S83RAYPLT	
TMATRD-CSH.U	- reads CSH travel times onto disc for use with S83RAYPLT	
DISTCOP.U	- copies inputted distances to the file GPT9.XMAT-CSSP	
DISTREAD.U	- reads distances from the file GPT9.XMAT-CSSP	
DISTCOPM25.U	- copies inputted distances to the file GPT9.XMAT-M25	
DISTRDM25B	- reads distances from the file GPT9.XMAT-M25	
TTREG.U	- performs regression analysis on PcP, PmP and Pn along CST, CSH CDS and CMP lines	
TTCORR.U	- corrects travel times for shallow structure down to a datum	
TTERMWAR.U	- performs analysis of WAR tt to yield average velocities and two-way trave times at each CMP position	
TTERMC.U	- time-term programme for Pn	
TTERMWAR.UN	- performs analysis of WAR tt to yield over two-way trave times at each CMP position	
CMPSYNSEI	- control programme for calls to SYNSEI	
SUBROUTINES	- subroutines required by the above programmes	
SEI583	- implementation - short notes	
*FS	- list of which sections are held on the *FS tapes	

```
C
CA     AHJ LEWIS
CD+
C************************************************************************
C                    PROGRAMME TR-INPUT
C************************************************************************
C
C  THIS PROGRAMME READS THE INFORMATION MATRIX GPT9:CSSP-TAPES WHICH HOLDS
C  THE LOCATION OF ALL CSSP TRACES ON THE CSSP DSDTF TAPES. UNITS ARE
C  ASSIGNED AS FOLLOWS:
C
C     UNIT 1 = GPT9:XMAT-M25  - FILE CONTAINING DIST'S FROM M25B
C     UNIT 2 = GPT9:XMAT-CSSP - DISTANCE MATRIX FOR CSSP
C     UNIT 3 = GPT9:CSSP-SHOT - FILE CONTAINING CSSP SHOT TIMES
C     UNIT 4 = GPT9:CSSP-TAPES- INFORMATION MATRIX FOR CSSP
C              UNITS 1 TO 4 ARE ASSIGNED INTERNALLY. THE LISTED FILES
C              MUST BE AVAILABLE ON GPT9 TO RUN THIS PROGRAMME
C     UNIT 5 )
C            ) = DEFAULT TO SCREEN
C     UNIT 6 )
C
C     UNIT 7 = OUTPUT OF TAPEREAD FILE IN FORMAT READY FOR USE AS INPUT TO
C              TR-CSSP
C     UNIT 8 = OUPUT OF SOURCE FILE WHICH CONTAINS TAPE MOUNT COMMANDS AND
C              RUN COMMANDS TO TAPERREAD THE REQUESTED TRACES OFF THE TAPES.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-
      INTEGER TNO, FNO, TOTFIL, ASTAT(110), ASHOT(110)
      REAL JDUR
      DIMENSION ISDAY(110), ISHOUR(110), ISMIN(110), SSEC(110),
     1          TITLES(20), TNO(110,110), FNO(110,110), IST(110),
     2          ISH(110), ISTAT(110), ISHOT(110), JSHOT(110), IN(110),
     3          DIST(110,110), KSTH(180), DIST25(180)
      DATA NSEA /'N   '/
      DATA ISEA /'M   '/
C
C
C     READ IN VALUES  IN MATRIX FILE
C
      ICDS=0
      ICMP=0
      CALL FTNCMD('ASSIGN 1=GPT9:XMAT-M25B', 23)
      CALL FTNCMD('ASSIGN 2=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 3=GLL4:CSSP-SHOT', 23)
      CALL FTNCMD('ASSIGN 4=GPT9:CSSP-TAPES', 24)
      READ (3,20) TITLES
      READ (3,20) TITLES
      READ (4,20) TITLES
   10 FORMAT (20A4)
      READ (4,20) (ISTAT(I),I=1,110)
      READ (4,20) (ISHOT(I),I=1,71)
   20 FORMAT (110A4)
C
      WRITE (6,30)
   30 FORMAT (/'READING MATRIX FILES')
      DO 60 I = 1, 110
        READ (4,40) (TNO(I,J),FNO(I,J),J=1,71)
   40   FORMAT (71(2I5))
        IF (I .GT. 71) GO TO 60
        READ (3,50) JSHOT(I), ISYEAR, ISDAY(I), ISHOUR(I), ISMIN(I),
     1  SSEC(I)
   50   FORMAT (2X, A4, 4X, 4I5, F10.3)
   60 CONTINUE
      MEM = 0
      DO 90 I = 1, 71
        IF (JSHOT(I) .EQ. ISHOT(I)) GO TO 90
   70   WRITE (6,80) JSHOT(I), ISHOT(I)
   80   FORMAT (/'SHOT IN TIME LIST AND INFO MATRIX ARE NOT THE SAME ',
     1         2A4)
        MEM = 1
   90 CONTINUE
      IF (MEM .EQ. 1) STOP
C
C
      ILINO = 1
  100 WRITE (6,110)
  110 FORMAT (/'COMMON STATION =1 OR COMMON SHOT =2 OR COMMON DIST= 3
     1 '/'OR COMMON MID POINT= 4 ?'/'_')
      READ (5,120) KIND
  120 FORMAT (I1)
  130 FORMAT (///'----------------------------------------------------'//
     1       )
      ISWIT = 1
      IF (KIND .EQ. 1) GO TO 140
      IF (KIND .EQ. 2) GO TO 310
      IF (KIND .EQ. 3) GO TO 410
      IF (KIND .EQ. 4) GO TO 560
C
CCCCCCC
C
C     COMMON STATION
C
CCCCCC
C
  140 WRITE (6,150)
  150 FORMAT (/'WHICH STATION NUMBER ?'/'__')
      READ (5,200) NSTAT
      NS = 0
      DO 160 I = 1, 110
  160 IF (NSTAT .EQ. ISTAT(I)) NS = I
      IF (NS .NE. 0) GO TO 180
      WRITE (6,170)
  170 FORMAT (/'COULD NOT FIND STAT GIVEN IN INFO. MATRIX !!')
      GO TO 710
  180 NSTAT = NS
      WRITE (6,190)
  190 FORMAT (/'NORTH OR IRISH SEA? (N OR M)')
```

```
      READ (5,200) ISEA
  200 FORMAT (A4)
      IF (ISEA .EQ. NSEA) GO TO 210
      NSTRT = 1
      TOTFIL = 36
      NEND = NSTRT + TOTFIL - 1
      GO TO 220
  210 NSTRT = 37
      TOTFIL = 30
      NEND = NSTRT + TOTFIL - 1
  220 WRITE (6,230)
  230 FORMAT (/'WHAT LENGTH OF DATA TO TAPEREAD IN SEC? F10.4')
      READ (5,240) JDUR
  240 FORMAT (F10.4)
      WRITE (7,250) TNO(NSTAT,NSTRT)
  250 FORMAT (10X, '*****************************************************
     1* **'/                    CALEDONIAN SUTURE SEISMIC PROJECT'/, 10X,
     2        '*****************************************************  **'/
     3        '                                TAPE', I5)
      WRITE (7,260) ISTAT(NSTAT), ISEA, ISWIT, TOTFIL
  260 FORMAT ('CST', A4, '/', A4, 8X, 2I5)
      WRITE (7,270)
  270 FORMAT (' FILE     SHOT    STATION          TIME ')
      K = TNO(NSTAT,NSTRT)
      I0 = 0
      IK = K
      IF (K .GT. 10) I0 = 1
      IF (K .GT. 10) IK = K - 10
      IF (K .GT. 20) I0 = 2
      IF (K .GT. 20) IK = K - 20
      WRITE (8,280) I0, IK, ILINO
  280 FORMAT ('$MOUNT GPT9:CSP', 2I1, 'B *T* NV RING=OUT FMT=FB(2048,204
     18)'/'$RUN GPT9:TR-CSP.COM+DURH:TRANSLIB 4=*T* 5=TR-IN(', I3,
     2        ') 7=GPT9:AHJL.DAT 2=-2(LAST+1)'/'$RELEASE *T*')
      L = 0
      DO 300 I = NSTRT, NEND
        IF (FNO(NSTAT,I) .EQ. 0) GO TO 300
        L = L + 1
        WRITE (7,290) FNO(NSTAT,I), ISHOT(I), ISTAT(NSTAT), ISYEAR,
     1  ISDAY(I), ISHOUR(I), ISMIN(I), SSEC(I), JDUR
  290   FORMAT (I6, 5X, A4, 6X, A4, 4X, 2(I4,2X), I2, 2X, I2, F8.4,
     1          F8.2)
  300 CONTINUE
      ILINO = L + 6 + ILINO
      GO TO 710
C
CCCCCCC
C
C     COMMON SHOT
C
CCCCCCC
C
  310 WRITE (6,320)
  320 FORMAT (/'WHICH SHOT ? - A4'/'____')
      READ (5,200) NSHOT
      WRITE (6,230)
      READ (6,240) JDUR
      NS = 0
      DO 330 I = 1, 71
  330 IF (NSHOT .EQ. ISHOT(I)) NS = I
      IF (NS .NE. 0) GO TO 350
      WRITE (6,340)
  340 FORMAT (/'COULD NOT FIND SHOT GIVEN IN INFO. MATRIX !!')
      GO TO 710
  350 NSHOT = NS
      DO 400 K = 1, 22
        L = 0
        DO 360 I = 1, 110
          IF (TNO(I,NS) .NE. K) GO TO 360
          L = L + 1
          IN(L) = I
  360   CONTINUE
        IF (L .EQ. 0) GO TO 400
        WRITE (7,250) K
        WRITE (7,370) ISHOT(NS), ISWIT, L
  370   FORMAT ('CSH/', A4, 12X, 2I5)
        WRITE (7,270)
        I0 = 0
        IK = K
        IF (K .GE. 10) I0 = 1
        IF (K .GE. 10) IK = K - 10
        IF (K .GE. 20) I0 = 2
        IF (K .GE. 20) IK = K - 20
        WRITE (8,380) I0, IK, ILINO
  380   FORMAT ('$MOUNT GPT9:CSP', 2I1, 'B *T* NV RING=OUT FMT=FB(2048,2
     1048)'/'$RUN GPT9:TR-CSP.COM+DURH:TRANSLIB 4=*T* 5=TR-CSH(', I3,
     2        ') 7=GPT9:CSH(LAST+1) 2=-2(LAST+1)'/'$RELEASE *T*')
        ILINO = 6 + L + ILINO
        DO 390 I = 1, L
          J = IN(I)
          IF (FNO(J,NS) .EQ. 0) GO TO 400
          WRITE (7,290) FNO(J,NS), ISHOT(NS), ISTAT(J), ISYEAR,
     1    ISDAY(NS), ISHOUR(NS), ISMIN(NS), SSEC(NS), JDUR
  390   CONTINUE
  400 CONTINUE
      GO TO 710
C
CCCCCCC
C
C     COMMON DISTANCE
C
CCCCCCC
C
  410 WRITE (6,420)
  420 FORMAT (/'WHICH DISTANCE ?'/'ALWAYS ENSURE DISTANCE GIVEN IS +/- 1
     1 KM OF AN ACTUAL DIST IN THE DATA F10.3'/'__________')
      READ (5,430) X
  430 FORMAT (F10.3)
      IX = X
      WRITE (6,230)
```

```
      READ (6,240) JDUR
      IF(ICDS.NE.0) GO TO 491
      READ (2,20) TITLES
      READ (2,20) (ASTAT(I),I=1,110)
      READ (2,20) (ASHOT(I),I=1,71)
      MEM = 0
      DO 470 I = 1, 110
        IF (ASTAT(I) .EQ. ISTAT(I)) GO TO 450
        WRITE (6,440)
440     FORMAT (/'STATS DO NOT CORRESPOND ON UNIT 2 & 4')
        MEM = 1
450     IF (ASHOT(I) .EQ. ISHOT(I)) GO TO 470
        WRITE (6,460)
460     FORMAT (/'SHOTS DO NOT CORRESPOND ON UNIT 2 & 4')
        MEM = 1
470   CONTINUE
      IF (MEM .EQ. 1) STOP
      WRITE (6,30)
      DO 490 I = 1, 110
        READ (2,480) (DIST(I,J),J=1,71)
480     FORMAT (71F10.0)
490   CONTINUE
491   ICDS=1
      XU = X + 1
      XD = X - 1
      L = 0
      DO 500 I = 1, 110
        DO 500 J = 1, 71
          IF (DIST(I,J) .GT. XU .OR. DIST(I,J) .LT. XD) GO TO 500
          L = L + 1
          IST(L) = I
          ISH(L) = J
500   CONTINUE
      NO = L
      DO 550 K = 1, 22
        L = 0
        DO 510 I = 1, NO
          M = IST(I)
          J = ISH(I)
          IF (TNO(M,J) .NE. K) GO TO 510
          L = L + 1
          IN(L) = I
510     CONTINUE
        IF (L .EQ. 0) GO TO 550
        WRITE (7,250) K
        WRITE (7,520) IX, ISWIT, L
520     FORMAT ('CDS/', I4, 12X, 2I5)
        WRITE (7,270)
        I0 = 0
        IK = K
        IF (K .GE. 10) I0 = 1
        IF (K .GE  10) IK = K - 10
        IF (K .GE. 20) I0 = 2
        IF (K .GE. 20) IK = K - 20
        WRITE (8,530) I0, IK, ILINO
530     FORMAT ('$MOUNT GPT9:CSP', 2I1, 'B *T* NV RING=OUT FMT=FB(2048,2
     1048)'/'$RUN GPT9:TR-CSP.COM+DURH:TRANSLIB 4=*T* 5=TR-CDS(', I3,
     2          ') 7=GPT9:CDS(LAST+1) 2=-2(LAST+1)'/'$RELEASE *T*')
        ILINO = 6 + L + ILINO
        DO 540 I = 1, L
          IK = IN(I)
          M = IST(IK)
          J = ISH(IK)
          IF (FNO(M,J) .EQ. 0) GO TO 540
          WRITE (7,290) FNO(M,J), ISHOT(J), ISTAT(M), ISYEAR, ISDAY(J),
     1    ISHOUR(J), ISMIN(J), SSEC(J), JDUR
540     CONTINUE
550   CONTINUE
      GO TO 710
C
CCCCCCCCCCCC
C
C       COMMON MID-POINT
C
CCCCCCCCCCCC
C
560   WRITE (6,570)
570   FORMAT (/'WHICH OMP ?'/'GIVE STATION AND SHOT WITH SMALLEST OFFSET
     1FOR THIS PARTICULAR OMP'/'____/____')
      READ (5,580) NSTAT, NSHOT
580   FORMAT (A4, 1X, A4)
      WRITE (6,230)
      READ (6,240) JDUR
      IF(IOMP.NE.0) GO TO 601
      WRITE (6,30)
      READ (1,10) TITLES
      READ (1,590) (KSTH(I),I=1,180)
      READ (1,600) (DIST25(I),I=1,180)
590   FORMAT (3(60A4))
600   FORMAT (3(60F10.4))
601   IOMP=1
      DO 610 I = 1, 180
        IF (NSTAT .EQ. KSTH(I)) NST = I
        IF (NSHOT .EQ. KSTH(I)) NSH = I
610   CONTINUE
      X = (DIST25(NST) + DIST25(NSH)) / 2.0
620   XU = X + 0.5
      XD = X - 0.5
      L = 0
      DO 650 I = 1, 180
        NSTRT = I + 1
        IF (I .EQ. 180) NSTRT = NSTRT - 1
        DO 650 J = NSTRT, 180
          XMP = (DIST25(I) + DIST25(J)) / 2.0
          IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 650
          L = L + 1
          IST(L) = 0
          ISH(L) = 0
          DO 640 IJ = 1, 110
            IF (KSTH(I) .EQ. ISTAT(IJ)) IST(L) = IJ
```

```
          IF (KSTH(J) .EQ. ISTAT(IJ)) IST(L) = IJ
 630      IF (IJ .GT. 71) GO TO 640
          IF (KSTH(J) .EQ. ISHOT(IJ)) ISH(L) = IJ
          IF (KSTH(I) .EQ. ISHOT(IJ)) ISH(L) = IJ
 640    CONTINUE
C
C        IF IST(L) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C        RESET L AND IST AND ISH
C
        IF (IST(L) .NE. 0 .AND. ISH(L) .NE. 0) GO TO 650
        IST(L) = 0
        ISH(L) = 0
        L = L - 1
 650 CONTINUE
     NO = L
     DO 700 K = 1, 22
       L = 0
       DO 660 I = 1, NO
         M = IST(I)
         J = ISH(I)
         IF (TNO(M,J) .NE. K) GO TO 660
         L = L + 1
         IN(L) = I
 660   CONTINUE
       IF (L .EQ. 0) GO TO 700
       WRITE (7,250) K
       WRITE (7,670) NSHOT, NSTAT, ISWIT, L
 670   FORMAT ('CMP', A4, '/', A4, 8X, 2I5)
       WRITE (7,270)
       I0 = 0
       IK = K
       IF (K .GE. 10) I0 = 1
       IF (K .GE. 10) IK = K - 10
       IF (K .GE. 20) I0 = 2
       IF (K .GE. 20) IK = K - 20
       WRITE (8,680) I0, IK, ILINO
 680   FORMAT ('$MOUNT GPT9:CSP', 2I1, 'B *T* NV RING=OUT FMT=FB(2048,2
    1048)'/'$RUN GPT9:TR-CSP.COM+DURH:TRANSLIB 4=*T* 5=TR-CMP(', I3,
    2          ') 7=GPT9:CMP(LAST+1) 2=-2(LAST+1)'/'$RELEASE *T*')
       ILINO = 6 + L + ILINO
       DO 690 I = 1, L
         IK = IN(I)
         M = IST(IK)
         J = ISH(IK)
         IF (FNO(M,J) .EQ. 0) GO TO 690
         WRITE (7,290) FNO(M,J), ISHOT(J), ISTAT(M), ISYEAR, ISDAY(J),
    1    ISHOUR(J), ISMIN(J), SSEC(J), JDUR
 690   CONTINUE
 700 CONTINUE
     GO TO 710
C
C
C
 710 WRITE (6,720)
 720 FORMAT (/'STOP (=0) OR CONTINUE (=1)  ?')
     READ (5,730) IQ
 730 FORMAT (I1)
     IF (IQ .EQ. 1) GO TO 100
     STOP
     END
C
```

```
C
CA    AHJ Lewis
CD+
C                          VERSION 14 OCT 1984
C        ********************************************************
C              CALEDONIAN SUTURE SEISMIC PROJECT
C        ********************************************************
C                          TAPEREAD
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    THIS  PROGRAMME COPIES INFORMATION FROM TAPES WRITTEN IN
C    DURHAM UNIVERSTY'S SEISMIC PROCESSING LAB ONTO MTS DISC SPACE.
C
C    UNITS ARE ASSIGNED AS FOLLOWS :-
C    3= TR-TCORRECT : IS ASSIGNED INTERNALLY AND CONTAINS STATION TT
C       TIME CORRECTIONS DUE TO ERRORS IN DIGITIZING THE TRACES FOR
C       THAT STATION : INCLUDES SUCH THINGS AS CORRECTION FOR TAPE-
C       HEAD MISALIGNMENTS ON THE RECORDING/PLAYBACK FACILITIES ETC.
C       FIRST LINE  CONTAINS NUMBER(NUM) OF STATIONS IN CORR FILE (I5)
C       NEXT NUM LINES CONTAIN STATION NAME ,ITS ORIGINAL ANALOGUE SOURCE
C       TAPE AND ITS CORRECTION IN SECS (A4,1X,F6.3)
C    4= TAPE
C    5= INPUT FILE
C    6= TERMINAL OUTPUT FOR UPDATE OF FILES BEING READ
C       THIS UNIT DEFAULTS TO THE SCREEN
C    7= OUTPUT FILE
C    8= A FILE CALLED BUFFER MUST BE AVAILABLE TO RUN THIS ROUTINE
C
C            EXAMPLE OF A RUN COMMAND
C    ____________________________________________
C    RUN *FTN SCARDS=TAPEREAD
C    RUN -LOAD+DURH:TRANSLIB 4=*T* 5=TESTP 7=-OUTPUT 2=SUMMARY
C    ____________________________________________
C
C    UNIT 6 APPEARS ON THE SCREEN
C    UNIT 8 IS USED INSIDE THE PROGRAMME
C    UNIT 2 IS SUMMARY OF FILES COPIED
C
C    TAPEREAD MUST BE COMPILED ON *FTN NOT *FTNX
C
C    DURH:TRANSLIB IS THE SYSTEM PROGRAMME WHICH CONTAINS THE
C                  SUBROUTINE TASEB WHICH CONVERTS ASCII TO EBDIIC
C
C    BYTE SWOPPING IS PERFORMED AS THE DATA IS WRITTEN BY DEC BUT
C    READ BY IBM MACHINES. BECAUSE OF FORTRAN LIMIT ONLY 32767 BYTES
C    CAN BE HELD IN AN ARRAY SO ONLY 16383 SAMPLES CAN BE TRANSFERRED.
C
C    SWIT=0    :   NUMBER OF SAMPLES IN EACH TRACE TO COPY
C        =1    :   LENGTH OF DATA TO COPY GIVEN IN SECONDS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-

      REAL*4 JSSEC,ISSEC,JDUR,KSSEC
      INTEGER TOTFIL,DAYDIF,HDIF,SWIT,ASTAT
      INTEGER*2 RBLK(16383),BLKTEM(1024),IBUFF(1024),LEM,LET
      LOGICAL*1 SAMP(2048),RSAMP(32766)
      EQUIVALENCE (BLKTEM,SAMP),(RBLK,RSAMP)
      DIMENSION TILE(18),FILTIT(18),SHEAD(5),TITLES(18),JSTAT(2),
     1 JEVENT(2),ASTAT(100),TCORR(100),JSORCE(100)
C
      COMMON/DSDTF/ SLAT,SLONG,SELEV,SAMRAT,RATERR,SSEC,DURAT,
     1 ELAT,ELONG,EELEV,ESEC
C
      COMMON/DSDTFI/ IBUFF,IVER,ID(3),IGYEAR,IGMON,IGDAY,IDATYP,
     1 IPROJ(4),IFILE,ISORCE(4),ISTAT(4),ICOMPS,IFLUT,ITIME,ISYEAR,
     2 ISDAY,ISHOUR,ISMIN,IBLKS,IETYPE(5),IEDAY,IEHOUR,IEMIN,IEVENT(5)
C
      COMMON/COMPSR/EIGEN(3),DAMP(3),GAIN(3),GENER(3),F1CORN(3),
     1 F1SLOP(3),F2CORN(3),F2SLOP(3),F3CORN(3),F3SLOP(3)
C
      COMMON/COMPSI/ICHAN(3),ICTYPE(3),IRECDR(6,3),ISEISM(6,3),ITRANS(3)
     1 ,ICOMP(3),IPOL(3),IF1TYP(2,3),IF2TYP(2,3),IF3TYP(2,3)
C
      DATA IAB/'AB  '/
C
C
C
C     HEADERS & SWITCHES FROM INPUT FILE READ IN
C
C
      REWIND 4
      CALL FTNOMD('ASSIGN 3=TR-TCORRECT;')
      READ(3,1) NUM
    1 FORMAT(I5)
      DO 6 I=1,NUM
        READ(3,2) ASTAT(I),JSORCE(I),TCORR(I)
    2   FORMAT(A4,1X,A4,F6.3)
    6   CONTINUE
      DO 5 I=1,3
        READ(5,7)(TILE(J),J=1,18)
    7   FORMAT(18A4)
        WRITE(7,8)(TILE(J),J=1,18)
        WRITE(2,8)(TILE(J),J=1,18)
    8   FORMAT(1X,18A4)
    5   CONTINUE
      READ(5,7)(FILTIT(J),J=1,18)
      READ(5,10)(SHEAD(J),J=1,5),SWIT,TOTFIL
   10 FORMAT(5A4,I5,I5)
      WRITE(7,20)(SHEAD(J),J=1,5),TOTFIL
      WRITE(2,20)(SHEAD(J),J=1,5),TOTFIL
   20 FORMAT(1X,5A4,I5)
      READ(5,7) TITLES
C
C
C
C     READ IN INPUT FILE AND READ TAPE FILE HEADERS(DSDTFR)
C
C
C
```

```
      KFILE=1
      DO 90 I=1,TOTFIL
        IF(SWIT.EQ.0) GO TO 40
        READ(5,30) JFILE,(JEVENT(J),J=1,2),(JSTAT(J),J=1,2),JSYEAR,JSDAY
     #  ,JSHOUR,JSMIN,JSSEC,JDUR
   30   FORMAT(I6,5X,2A4,2X,2A4,I4,2X,I4,2X,I2,2X,I2,F8 4,F8.2)
        GO TO 55
   40   READ(5,50) JFILE,(JEVENT(J),J=1,2),(JSTAT(J),J=1,2),JSYEAR,JSDAY
     #  ,JSHOUR,JSMIN,JSSEC,JSAMP
   50   FORMAT(I6,5X,2A4,2X,2A4,I4,2X,I4,2X,I2,2X,I2,F8.4,I6)
   55   NFILES=JFILE-KFILE
        NRCDS=0
        IF(NFILES.EQ.0.AND.JFILE.NE.1) GO TO 42
        IF(JFILE.EQ.1) REWIND4
        IF(JFILE.EQ.1) NFILES=0
        IF(NFILES.LT.0) GO TO 43
        CALL SKIP(NFILES,NRCDS,4,&1000,&1010,&1020)
        GO TO 45
   42   CALL SKIP(-1,NRCDS,4,&1000,&1010,&1020)
        CALL SKIP(1,NRCDS,4,&1000,&1010,&1020)
        GO TO 45
   43   NFILES=NFILES-1
        CALL SKIP(NFILES,NRCDS,4,&1000,&1010,&1020)
        CALL SKIP(1,NRCDS,4,&1000,&1010,&1020)
   45   CONTINUE
        CALL READ(IBUFF,LEM,0,LNUN,4,&1030)
        KFILE=JFILE
C
C
C        CONVERSION FROM ASCII TO EBDIIC AND BYTE SWOPPING
C        ARE PERFORMED BY THE NUMAC ROUTINE TASEB
C        DECODE DSDTFR HEADER & CHECK CORRECT FILE HAS BEEN READ
C
C
C
        LENT=2048
        CALL TASEB(IBUFF,LENT)
        CALL DSDTFR
        IF (JSDAY.LT.ISDAY) GO TO 1040
        DO 57 J=1,2
          IF(IEVENT(J).NE.JEVENT(J)) GO TO 1040
          IF(ISTAT(J).NE.JSTAT(J)) GO TO 1040
   57     CONTINUE
        IF(JSHOUR.LT.ISHOUR) GO TO 1040
        L=0
        DO 559 K=1,NUM
           IF(ISTAT(1).NE.ASTAT(K)) GO TO 559
           IF(JSORCE(K).EQ.IAB.OR.JSORCE(K).EQ.ISORCE(1)) GO TO 557
  559      CONTINUE
        GO TO 58
  557   L=K
        SSEC=SSEC+TCORR(L)
        IF(SSEC.LT.0.0) IEMIN=IEMIN-1
        IF(SSEC.LT.0.0) SSEC=60.0+SSEC
        WRITE(6,558) TCORR(L),ISORCE(1)
  558   FORMAT(/'NEXT TRACE IS CORRECTED BY ',F6.3,' SECONDS - SOURCE TA
     *PE IS ',A4)
   58   WRITE(6,999)(IEVENT(K),K=1,2),(ISTAT(K),K=1,2)
  999   FORMAT(1X,2A4,2A4)
C
C
C
C        CALCULATION OF TIME OF FIRST SAMPLE IN OUTPUT FILE & TOTAL
C        NUMBER OF SAMPLES TO READ FROM TAPE
C
C
C
        A1=JSDAY-ISDAY
        A2=JSHOUR-ISHOUR
        A3=JSMIN-ISMIN
        A4=JSSEC-SSEC
        TDIFF=((A1*24.0+A2)*60.0+A3)*60.0+A4
        IF(TDIFF.LT.0) GO TO 1060
        A5=SAMRAT*TDIFF
        JSAM=IFIX(A5)+1
   67   IF(SWIT.EQ.0) GO TO 60
        H=JDUR*SAMRAT
        JSAMP=IFIX(H)+1
        GO TO 61
   60   JDUR=(JSAMP-1)/SAMRAT
   61   IF(JSAMP.GT.16383) GO TO 1050
        JTSAMP=JSAM+JSAMP-1
        TIME=FLOAT(JSAM-1)/SAMRAT
        TTIME=TIME+JDUR
        IF(TTIME.GT.DURAT) GO TO 1046
   59   B=TIME/(60.0*60.0*24.0)
        DAYDIF=IFIX(B)
        TEMP=TIME-(FLOAT(DAYDIF)*60.0*60.0*24 0)
        C=TEMP/(60.0*60.0)
        HDIF=IFIX(C)
        TEMP=TEMP-(FLOAT(HDIF)*60.0*60.0)
        D=TEMP/60.0
        MINDIF=IFIX(D)
        SECDIF=TEMP-(MINDIF)*60.0
        EE=0
        FF=0
        GG=0
        KSSEC=SSEC+SECDIF
        IF(KSSEC.LT.60.0) GO TO 62
        KSSEC=KSSEC-60.0
        EE=1
   62   KSMIN=EE+ISMIN+MINDIF
        IF(KSMIN.LT.60) GO TO 64
        KSMIN=KSMIN-60
        FF=1
   64   KSHOUR=FF+ISHOUR+HDIF
        IF(KSHOUR.LT.24) GO TO 66
        KSHOUR=KSHOUR-24
        GG=1
   66   KSDAY=GG+ISDAY+DAYDIF
```

```
C
C
C
C       READ IN TRACE SAMPLES FROM TAPE
C       LEM RETURNS NUMBER OF BYTES READ
C       SAMPLES ARE TWO BYTES LONG
C       BYTE SWOPPING ON THE DATA IS PERFORMED
C
C
C
      LENTOT=0
      K=0
   69 LEM=0
      CALL READ(BLKTEM,LEM,0,LNUM,4,1030)
      LEN=LEM/2
      DO 70 J=1,LEN
         L=LENTOT+J
         IF(L.LT.JSAM) GO TO 70
         IF(L.GT.JTSAMP) GO TO 71
            K=K+1
            MR=2*K-1
            MT=2*J-1
            NR=2*K
            NT=2*J
            RSAMP(MR)=SAMP(NT)
            RSAMP(NR)=SAMP(MT)
   70    CONTINUE
      LENTOT=LEN+LENTOT
      IF(K.LT.JSAMP) GO TO 69
   71 IF(K.NE.JSAMP) GO TO 1048
C
C
C
C       WRITE HEADER AND TRACE TO OUTPUT FILE ONTO 2 LINES
C       ALSO CALCULATE DISTANCE AND AZIMUTH BETWEEN SHOT AND
C       RECEIVER USING THE SUBROUTINE DISTAZ.
C       LET IS NUMBER OF BYTES TO WRITE TO OUTPUT FILE
C       SAMPLES ARE 2 BYTES LONG
C
C
C
      IF(SLAT.NE.0.0.OR.SLONG.NE.0.0) GO TO 74
      DIST=5.529+DIST
      DAZ=0.0
      GO TO 75

   74 CALL DISTAZ(SLAT,SLONG,ELAT,ELONG,DIST,DAZ)
   75 WRITE(7,80)(IEVENT(J),J=1,2),ELAT,ELONG,EELEV,IEDAY,IEHOUR,IEMIN
    1 ,ESEC,(ISTAT(J),J=1,2),SLAT,SLONG,SELEV,KSDAY,KSHOUR,KSMIN,KSSEC
    2 ,JSAMP,SAMRAT,JDUR,GAIN(1),IPOL(1),DIST,DAZ
   80 FORMAT(2A4,2X,3F12.6,I4,1X,I2,2X,I2,F8.4,2X,2A4,2X,3F12.6,I4,1X,
    1 I2,2X,I2,F8.4,2X,I6,F10.4,F8.2,2X,E10.4,2X,A4,2F10.3)
C
      WRITE(2,81)(IEVENT(J),J=1,2),ELAT,ELONG,EELEV,IEDAY,IEHOUR,IEMIN
    1 ,ESEC,(ISTAT(J),J=1,2),SLAT,SLONG,SELEV,KSDAY,KSHOUR,KSMIN,KSSEC
    2 ,JSAMP,SAMRAT,JDUR,GAIN(1),IPOL(1),DIST,DAZ
   81 FORMAT(2A4,2X,3F12.6,I4,1X,I2,2X,I2,F8.4,/,2A4,2X,3F12.6,I4,1X,
    1 I2,2X,I2,F8.4,/,I6,F10.4,F8.2,2X,E10.4,2X,A4,2F10.3,/)
C
      LET=JSAMP*2
      CALL WRITE(RBLK,LET,0,LNUM,7,&1070)
   90 CONTINUE
      GO TO 156
C
C
C     ERROR MESSAGES
C
C
 1000 WRITE(6,1005)
 1005 FORMAT('EOF REACHED BEFORE SKIP COMPLETE')
      GO TO 90
 1010 WRITE(6,1015)
 1015 FORMAT('BOT REACHED ON TAPE REWIND')
      GO TO 90
 1020 WRITE(6,1025)
 1025 FORMAT('UNIT NUMBER IS ASSIGNED INCORRECTLY')
      GO TO 90
 1030 WRITE(6,1035)
 1035 FORMAT('EOF REACHED DURING READ OF TAPE')
 1040 WRITE(6,1045)
 1045 FORMAT('*********INPUT & TAPE INFO DO NOT MATCH IN NEXT READ******
     1***')
      GO TO 58
 1046 JTSAMP=IFIX(DURAT*SAMRAT)+1
      JSAMP=JTSAMP-JSAM+1
      WRITE(6,1047) JSAMP
 1047 FORMAT(1X,'NO. OF SAMPLES TO COPY OVER EXCEEDS NO. ON TAPE',/,'NO
     1. OF SAMPLES COPIED IS ',I5)
      GO TO 59
 1048 WRITE(6,1049)
 1049 FORMAT(1X,'NO. OF SAMPLES COPIED DO NOT EQUAL NO. REQUESTED')
      GO TO 90
 1050 WRITE(6,1055)
 1055 FORMAT(1X,'LINES CAN ONLY BE 32673 BYTES LONG IN OUTPUT')
      GO TO 90
 1060 WRITE(6,1065)
 1065 FORMAT(1X,'START TIME ON UNIT 6 IS BEFORE TIME OF FIRST SAMPLE'
     */'COPYING FROM FIRST SAMPLE')
      JSAM=1
      A5=0.0
      TDIFF=0.0
      GO TO 67
 1070 WRITE(6,1075)
 1075 FORMAT(1X,'ERROR IN DATA WRITE STATEMENT')
      GO TO 90
  156 REWIND 4
      STOP
      END
```

```
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE DISTAZ(AL,OL,BL,CL,DIST,DAZ)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     CALCULATES DISTANCE BETWEEN SHOT AND STATION FROM DECIMAL DEGREES
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     ORIGINAL VERSION G.K.WESTBROOK
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      DISTANCE CALCULATOR ACCURATE TO 1 IN 10**7 AT 1600 KM
C     USES FORMULA OF ROBBINS, A.R. 'LONG LINES ON THE
C     SPHEROID' EMP. SVY. REV. 125 1962. WRITTEN FOR
C     FINDING DISTANCES BETWEEN RECEIVING STATIONS AND SHOT
C     POINTS IN SEISMIC EXPERIMENTS, BUT CAN BE USED FOR ANY
C     DISTANCE CALCULATION. NLP = NO. OF STATIONS.  NSP = NO
C     OF SHOTS.  NL(I) = IDENTIFICATION NO. OF ITH RECEIVING
C     STATION.  AL= DEGREES OF LATITUDE.  AM = MINUTES OF LA
C     TITUDE.  OL = DEGREES OF LONGITUDE.  CM = MINUTES OF L
C     TITUDE.  OL = DEGREES OF LONGITUDE.  CM = MINUTES OF
C     LONGITUDE.  NS(I) = IDENTIFICATION NO. OF ITH SHOT.
C     LONGITUDE : EAST IS POSITIVE
C     LATITUDE : NORTH IS POSITIVE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION DL,HL,ERO,PRO,A,B,SS,VS,SI,V,CO,S,SDL,CDL,DIF,S1,
     1C1,G,GA,EC,EL,TA,GS,FL,GL
      DIMENSION NL(50),A(2,50),NS(500),B(2,500),SI(50),CO(50),V(50),VS(5
     100),SS(500),GA(50),TA(500),GS(500),TITLE(20),NT(500)
      DTR=0.01745329
      ER=6378.160
      PR=6356.775
      ERO=ER**2
      PRO=PR**2
      EC=(ERO-PRO)/ERO
      EL=(ERO-PRO)/PRO
      DL=AL*DTR
      HL=OL*DTR
      SS(1)=DSIN(DL)
      TA(1)=SS(1)/DCOS(DL)
      VS(1)=ER/DSQRT(1-EC*SS(1)**2)
      GS(1)=DSQRT(EL*SS(1)**2)
      FL=BL*DTR
      GL=CL*DTR
      B(1,2)=FL
      B(2,2)=GL
      S1=DSIN(FL)
      C1=DCOS(FL)
      V1=ER/DSQRT(1-EC*S1**2)
      DIF=HL-GL
      CDL=DCOS(DIF)
      SDL=DSIN(DIF)
      SS(2)=S1
      VS(2)=V1
      TA(2)=S1/C1
      GS(2)=DSQRT(EL*S1**2)
      G=GS(2)
      T=(1.0-EC+EC*V1*S1/(VS(1)*SS(1)))*TA(1)
      AT=ATAN(T)
      PAZ=C1*T-S1*CDL
      SL=SDL
      AZ=ATAN2(SL,PAZ)
      HQ=EL*C1**2*COS(AZ)**2
      H=SQRT(HQ)
      BAZ=ABS(AZ)-3.14159
      IF(BAZ.GT.-1.0.OR.BAZ.LT.-2.14159) GO TO 57
      FD=SDL*COS(AT)/SIN(AZ)
      D=ARSIN(FD)
      GO TO 58
   57 D=ARSIN(COS(AT)*PAZ/COS(AZ))
   58 DIST=V1*D*(1.0-D**2*HQ*(1.0-HQ)/6.0+D**3*G*H*(1.0-2.0*HQ)/8.0+D**4
     1*(HQ*(4.0-7.0*HQ)-3.0*G**2*(1.0-7.0*HQ))/120.0-D**5*G*H/48.0)
      DAZ=AZ/DTR
      RETURN
      END


C$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE DSDTFR
C$$$$$$$$$$$$$$$$$$$$$$$
C
C     DECODES CSSP HEADER BLOCKS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE TO LOAD VARIABLES FROM HEADER BUFFER
C DATA IS OUTPUT TO COMMON BLOCKS
C THIS IS THE CONVERTED NUMAC VERSION
C BLOCK DSDTF&DSDTFI CONTAIN EVENT AND RECORD INFORMATION
C OTHER BLOCKS CONTAIN COMPONENT INFORMATION
C DATA IS IN I*2 FORMAT IN IBUFF : INTEGERS ARE CONVERTED TO I*4
C (THE NUMAC DEFAULT) USING ASSIGN STATEMENTS WHILST ALPHNUMERIC
C I*2 CHARACTERS ARE CONVERTED TO I*4 USING EQUIVALENCE
C STATEMENTS - ALL VALUES ARE RETURNED VIA THE COMMON BLOCKS AS
C I*4 AND R*4(THE NUMAC DEFAULT).
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C ORIGINAL PROGRAMME D.L. STEVENSON
C VERSION ON NUMAC   A.H.J.LEWIS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT INTEGER*2(J)
      INTEGER*2 IBUFF(1024)

      INTEGER*2 JD(3),JPROJ(4),JSORCE(4),JSTAT(4),JEVENT(5),JETYPE(5),
     1 JCHAN(3),JCTYPE(3),JRECDR(6,3),JSEISM(6,3),JTRANS(3),
     2 JCOMP(3),JPOL(3),JF1TYP(2,3),JF2TYP(2,3),JF3TYP(2,3)
```

```
C
      EQUIVALENCE (JD,ID),(JPROJ,IPROJ),(JSORCE,ISORCE),
     1 (JSTAT,ISTAT),(JEVENT,IEVENT),(JETYPE,IETYPE),
     2 (JCHAN,ICHAN),(JCTYP,ICTYP),(JRECDR,IRECDR),
     3 (JSEISM,ISEISM),(JTRANS,ITRANS),(JCOMP,ICOMP),(JPOL,IPOL),
     4 (JF1TYP,IF1TYP),(JF2TYP,IF2TYP),(JF3TYP,IF3TYP)

      COMMON/DSDTF/ SLAT,SLONG,SELEV,SAMRAT,RATERR,SSEC,DURAT,
     1 ELAT,ELONG,EELEV,ESEC
C
      COMMON/DSDTFI/ IBUFF,IVER,ID(3),IGYEAR,IGMON,IGDAY,IDATYP,
     1 IPROJ(4),IFILE,ISORCE(4),ISTAT(4),ICOMPS,IFLUT,ITIME,ISYEAR,
     2 ISDAY,ISHOUR,ISMIN,IBLKS,IETYPE(5),IEDAY,IEHOUR,IEMIN,IEVENT(5)
C
      COMMON/COMPSR/EIGEN(3),DAMP(3),GAIN(3),GENER(3),F1CORN(3),
     1 F1SLOP(3),F2CORN(3),F2SLOP(3),F3CORN(3),F3SLOP(3)
C
      COMMON/COMPSI/ICHAN(3),ICTYPE(3),IRECDR(6,3),ISEISM(6,3),ITRANS(3)
     1 ,ICOMP(3),IPOL(3),IF1TYP(2,3),IF2TYP(2,3),IF3TYP(2,3)

C DECODE DATA

      CALL ATTEMP
      REWIND 8
      WRITE(8,100) IBUFF
  100 FORMAT(16(64A2))
      REWIND 8
      READ(8,110,END=99) JVER,(JD(J),J=1,3),JGYEAR,JGMON,JGDAY,JDATYP,
     1 (JPROJ(J),J=1,4),JFILE,(JSORCE(J),J=1,4),(JSTAT(J),J=1,4),SLAT,
     2 SLONG,SELEV,JCOMPS,SAMRAT,RATERR,JFLUT,JTIME,JSYEAR,JSDAY,JSHOUR,
     3 JSMIN,SSEC,DURAT,JBLKS,(JEVENT(J),J=1,5),(JETYPE(J),J=1,5),ELAT,
     4 ELONG,EELEV,JEDAY,JEHOUR,JEMIN,ESEC,JCHAN(1),JCTYPE(1),
     5 (JRECDR(J,1),J=1,6),(JSEISM(J,1),J=1,6),JTRANS(1),JCOMP(1),JPOL(1
     6 ),EIGEN(1),DAMP(1),GAIN(1),GENER(1),(JF1TYP(J,1),J=1,2),F1CORN(1)
     7 ,F1SLOP(1),(JF2TYP(J,1),J=1,2),F2CORN(1),F2SLOP(1),(JF3TYP(J,1),
     8 J=1,2),F3CORN(1),F3SLOP(1),JCHAN(2),JCTYPE(2),(JRECDR(J,2),J=1,6)
     9 ,(JSEISM(J,2),J=1,6),JTRANS(2),JCOMP(2),JPOL(2),EIGEN(2),DAMP(2),
     A GAIN(2),GENER(2),(JF1TYP(J,2),J=1,2),F1CORN(2),F1SLOP(2),
     B (JF2TYP(J,2),J=1,2),F2CORN(2),F2SLOP(2),(JF3TYP(J,2),J=1,2),
     C F3CORN(2),F3SLOP(2),JCHAN(3),JCTYPE(3),(JRECDR(J,3),J=1,6),
     D (JSEISM(J,3),J=1,6),JTRANS(3),JCOMP(3),JPOL(3),EIGEN(3),DAMP(3),
     E GAIN(3),GENER(3),(JF1TYP(J,3),J=1,2),F1CORN(3),F1SLOP(3),
     G (JF2TYP(J,3),J=1,2),F2CORN(3),F2SLOP(3),(JF3TYP(J,3),J=1,2),
     H F3CORN(3),F3SLOP(3)
  110 FORMAT(14X,I2,26X,3A2,10X,3I2,10X,I2,12X,4A2,6X,I6,12X,4A2,24X,
     #4A2,3(4X,F12.6),12X,I2,2X,2(6X,F10.4),8X,I2,4X,I2,12X,2I4,2I2,F8.4
     #,8X,F8.2,12X,I4,16X,2(6X,5A2),3(4X,F12.6),16X,I4,2I2,F8.4,16X,16X,
     #16X,3(8X,I2,4X,I2,2(4X,6A2),12X,A2,14X,A2,2X,4X,A2,4X,F6.2,8X,F8.2
     #,6X,E10.4,6X,E10.4,3(16X,12X,2I2,6X,E10.4,6X,E10.4),16X,16X,16X,16
     #X,16X,16X,16X,16X,16X,16X,16X))
      IVER=JVER
      IGYEAR=JGYEAR
      IGDAY=JGDAY
      IDATYP=JDATYP
      IFILE=JFILE
      ICOMPS=JCOMPS
      IFLUT=JFLUT
      ITIME=JTIME
      ISYEAR=JSYEAR
      ISDAY=JSDAY
      ISHOUR=JSHOUR
      ISMIN=JSMIN
      IBLKS=JBLKS
      IEDAY=JEDAY
      IEHOUR=JEHOUR
      IEMIN=JEMIN
   99 RETURN
      END


C$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE ATTEMP
C$$$$$$$$$$$$$$$$$$$$$$$
C
C     THIS SUBROUTINE HOLDS THE SYSTEM DEPENDENT ROUTINE WHICH
C     CREATES A BUFFER FILE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL FTNCMD('ASSIGN 8=-BUFFER',16)
      RETURN
      END
C
```

```
C
CA    AHJ LEWIS
CD+
C
C     PROGRAMME TR-IGS
C     ____________
C
C     THIS PROGRAMME READS THE DIGITAL TAPES PRODUCED AT THE EDINBURGH FACILITY
C     AT THE GLOBAL SEISMOLOGY UNIT. THE TAPES ARE ALL HELD ON GPT9 AND
C     LABELLED GPT9:G8310A TO GPT9:G8320A.
C
C     THE PROGRAMME DECODES THE TIME CODE USING ROUTINES WRITTEN BY DAVE
C     STEVENSON. THESE ROUTINES HAVE SINCE BEEN IMPROVED.
C     ONCE THE SAMPLING RATE AND EVENT IS LOCATED IT IS COPIED TO DISC
C     IN THE FORMAT REQUIRED BY THE PLOTTING PROGRAMME CSSPLOT.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  ISAM-ARRAY CONTAINING DEMULTIPLEXED DATA
C  IST-ELEMENT IN ARRAY WHERE START WRITING TO DISC
C  ISTSAM-ELEMENT IN ARRAY TO COPY FROM DISC USING GIVEN TIME
C  ISTORE=NO. OF ELEMENTS FROM STARTING PT.THAT REQUIRE WRITING TO DISC
C  NOSAM-NO. OF SAMPLES IN PARTICULAR PART OF ARRAY
C  BLK,SAMP-ARRAYS CONTAINING DATA READ FROM 1 BLOCK IN FILE
C  RBLK,RSAMP-ARRAYS CONTAINING ABOVE DATA AFTER BYTE SWAP
C  INFO-HEADER INFO. FOR EACH FILE READ
C  LEN,LNUM-REFER TO SUBROUTINE READ & WRITE
C  MEM     : SET TO 0 OR 11 IF ON SECOND PASS THROUGH PROGRAMME
C  MEMS    : SET TO 1 DURING LOOPS THROUGH DECODE WHEN LOCATING
C            NEAREST MINUTE TO EVENT TIME
C  ISTMEM  : LOCATES ISTSAM (FROM SRTSAM) IN TRUNCATED CLOCK ARRAY ICLK
C            TO ITS ORIGINAL SAMPLE NO. IN ISAM DATA ARRAY
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  UNIT 4   :  TAPE
C  UNIT 5   :  INPUT FILE
C  UNIT 6   :  RECORD OF RUN OUTPUT HERE - ERROR MESSAGES
C                                          TIME DECODING DETAILS
C  UNIT 7   :  OUTPUT ONTO DISC OF FINAL DATA
C  UNIT 8   :  OUTPUT OF TRACE HEADERS WRITTEN TO UNIT 7 - SERVES AS
C              A RECORD OF THE DEMULTIPLEXED TRACES
C
CCCCC
C
C  $MOUNT TAPENAME *T* NV RING=OUT FMT=FB(4096,4096)
C  RUN *FTNX SCARDS=TAPEREAD.IGS
C  RUN -LOAD+TRURAT.COM+DECODE.COM 4=*T* 5=INPUT 6=RUN OUTPUT 7=DATA
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ORIGINAL PROGRAMME TPTODSC.G                              TIM SUMMERS
C ORIGINAL TRURAT & DECODE SUBROUTINES                      DAVE STEVENSON
C MODIFICATIONS TO PRODUCE TAPEREAD FORMAT               )
C                TO ENCORPORAT TIME DECODING ROUTINES )  ANTHONY LEWIS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-
C
      DIMENSION TILE(18), FILTIT(18), TITLES(18), SHEAD(5), DAZ(16)
      DIMENSION SLAT(16), SLONG(16), SELEV(16)
      DIMENSION ISTOR(16), INFO(19), JSTAT(16), IST(16), DIST(16)
      INTEGER LNUM, BLOCK, SRTSAM, BLKMAX, TOTFIL, SWIT, SWITS
      INTEGER*2 RBLK(2048), BLK(2048), ISAM(320000), LEN, ICLK(40000)
      INTEGER*2 SPEC, LERR, RESET, ERROR, YEAR, MONTH, DATE, DAY,
     1          KSHOUR, KSMIN, JULDAY
      REAL JDUR, ISSEC
      LOGICAL*1 SAMP(4096), RSAMP(4096)
      EQUIVALENCE (BLK(1),SAMP(1)), (RBLK(1),RSAMP(1))
      DATA IST(1) /1/, IST(2) /20000/, IST(3) /40000/, IST(4) /60000/,
     1     IST(5) /80000/, IST(6) /100000/, IST(7) /120000/,
     2     IST(8) /140000/, IST(9) /160000/, IST(10) /180000/, IST(11)
     3     /200000/, IST(12) /220000/, IST(13) /240000/, IST(14) /
     4     260000/, IST(15) /280000/, IST(16) /300000/
C
C
C READ HEADER OF INPUT FILE AND WRITE TO OUTPUT FILE PLUS -
C ENTER NO.OF FILES TO BE TRANSFERED AND NO. OF CHANNELS PER FILE
C
C
      REWIND 4
      NFILE = 0
      DO 30 I = 1, 3
        READ (5,10) (TILE(J),J=1,18)
   10   FORMAT (18A4)
        WRITE (8,20) TILE
        WRITE (7,20) TILE
   20   FORMAT (1X, 18A4)
   30 CONTINUE
      READ (5,10) (FILTIT(J),J=1,18)
      WRITE (6,40)
   40 FORMAT ('ENTER DATA HEADER & TOTFIL & NCHAN & SWIT '/
     1   '________ ______.....______.....')
      READ (5,50) (SHEAD(J),J=1,5), TOTFIL, NCHAN, SWIT, SWITS
   50 FORMAT (5A4, 4I5)
      WRITE (8,60) (SHEAD(J),J=1,5), TOTFIL, NCHAN, SWIT, SWITS
      WRITE (7,60) (SHEAD(J),J=1,5), TOTFIL, NCHAN, SWIT, SWITS
   60 FORMAT (1X, 5A4, 4I5)
C
CCCCCC
C
C
C      INPUT CHANNEL ASSOCIATIONS FOR THE DIGITAL TAPE TO BE USED
C      INPUT DISTANCES OF STATIONS FROM CENTRE OF ARRAY FOR ARRAY
C      DATA
C
C
CCCCCC
C
      WRITE (6,70)
```

```
   70 FORMAT ('DO YOU WANT CLOCKS COPIED OVER'/'BOTH NSTRT=1 : MSF ONLY
     1 NSTRT=2 : NONE NSTRT=2(OR 3 IF MSF IS PRESENT)'/'—')
      READ (5,120) NSTRT
CCCCC
C
      IF (SWIT .EQ. 1) GO TO 80
C
CCCCC
      READ (5,130) (JSTAT(J),J=1,16)
      READ (5,140) (DIST(J),J=1,8)
      READ (5,140) (DAZ(J),J=1,8)
      READ (5,140) (DIST(J),J=9,16)
      READ (5,140) (DAZ(J),J=9,16)
C
      GO TO 110
C
   80 DO 100 I = 1, NCHAN
         READ (5,90) JSTAT(I), SLAT(I), SLONG(I), SELEV(J)
   90    FORMAT (A4, 3F12.6)
  100 CONTINUE
C
  110 READ (5,10) TITLES
  120 FORMAT (I2)
  130 FORMAT (16A4)
  140 FORMAT (8F10.4)
C
C
C
C
C*****************************************************************
C
C
C                 START LOOP FOR A NUMBER OF FILES
C                 ________________________________
C
C*****************************************************************
C
C
C
C
      KFILE = 1
      BLKMAX = 20000 * NCHAN / 2048
      ITEMP = BLKMAX - 1
      DO 790 II = 1, TOTFIL
C
CCCCCC
C
C     DATA DIGITIZED AT EDINBURGH HAS INTERNAL CLOCK ON CHANNEL 1 AND
C     MSF ON CHANNEL 2 IF PRESENT : NEED TO INPUT WETHER MSF IS
C     AVAILABLE
C
CCCCCC
C
  150   WRITE (6,160)
  160   FORMAT (//'FILE NUM., INPUT WHICH CHANNEL TO READ CLOCK FROM?'/
     1          '   IF IS 2 THEN',
     2          ' PROGRAMME WILL ASSUME IT IS USING MSF'/
     3
     4' AND USE THE RELEVENT   SWITCHES IN THE DECODE SUBROUTINE'/
     5
     6'                                ALSO INPUT APPROX. SAMPLE RATE'/
     7          '(A NEGATIVE VALUE MEANS SAMPLE RATE GIVEN IS ACCURATE'/
     8          'FILE NO,CLOCK CHANNEL,CLK TYPE,SAMPLE RATE '/
     9          '———..—..... —...')
      NRCDS = 0
      READ (5,170) JFILE, NRCDS, JCLK, KCLK, SRAT
  170 FORMAT (2I5, 2I2, F10.5)
C
CCCCCC
C
      WRITE (6,20) (TILE(J),J=1,18)
      WRITE (6,180) JFILE
  180 FORMAT (//'BYTE SWAPPING AND DEMULTIPLEXING DATA'/
     1          'FILE NUMBER ', I5, //)
      WRITE (6,20) (TILE(J),J=1,18)
C
CCCCCC
C
      IF (NBLOCK .GT. 0) KFILE = KFILE + 1
      MEM = 0
      NBLOCK = 0
      NFILES = JFILE - KFILE
      NRDS = 0
      IF (JFILE .EQ. 1) REWIND 4
      IF (JFILE .EQ. 1) NFILES = 0
      IF (NFILES .EQ. 0 .AND. JFILE .NE. 1) GO TO 190
      IF (NFILES .LT. 0) GO TO 200
      CALL SKIP(NFILES, NRCDS, 4, &980, &1000, &1020)
      GO TO 210
  190 CALL SKIP(-1, NRDS, 4, &980, &1000, &1020)
      CALL SKIP(1, NRCDS, 4, &980, &1000, &1020)
      GO TO 210
  200 NFILES = NFILES - 1
      CALL SKIP(NFILES, NRDS, 4, &980, &1000, &1020)
      CALL SKIP(1, NRCDS, 4, &980, &1000, &1020)
      GO TO 210
  210 KFILE = JFILE

  220 DO 230 I = 1, 16
  230 ISTOR(I) = IST(I)
C
C
      CALL READ(BLK, LEN, 0, LNUM, 4, &800)
      BLOCK = 1
      LERR = 0
      GO TO 250
  240 BLOCK = BLOCK + 1
      CALL READ(BLK, LEN, 0, LNUM, 4, &370)
  250 IBLK = LEN / 2
      ICONT = LEN - 1
```

```
C
C
C     1 BLOCK OF DATA HAS NOW BEEN READ
C     PAIRS OF BYTES ARE NOW SWAPPED
C
C
 260   DO 270 I = 1, ICONT, 2
         CALL MOVEC(1, SAMP(I), RSAMP(I + 1))
         CALL MOVEC(1, SAMP(I + 1), RSAMP(I))
 270   CONTINUE
C
C
C     1 BLOCK NOW IN FORMAT TO READ ON MTS
C     DEMULTIPLEXING IS NOW PERFORMED
C
C
CCCCCC
C
C    CHECK WETHER ALL CHANNELS ARE TO BE DEMULTIPLEXED :
C    IF SO GO TO 26 : IF NOT JUST DEMULT. CLOCK CHANNEL.
C
CCCCCC
       IF (BLOCK .LT. BLKMAX) GO TO 300
       J = LENGTH
       DO 280 I = JCLK, IBLK, NCHAN
         J = J + 1
         ICLK(J) = RBLK(I)
         IF (J .EQ. 24000) GO TO 290
 280   CONTINUE
 290   LENGTH = J
       IF (LENGTH .EQ. 24000) GO TO 380
CCCCCC
C
C    NOW PROCEED TO DECODING ROUTINES IF ENOUGH CLOCK DEMULT.
C
CCCCC
C
C
       GO TO 240
C
C
CCCCC
C
C     DEMULT. ALL CHANNELS
C
CCCCC
 300   JCH = 0
       DO 330 I = 1, NCHAN
         JCH = JCH + 1
         JSAMP = ISTOR(I)
         DO 310 J = JCH, IBLK, NCHAN
           ISAM(JSAMP) = RBLK(J)
           JSAMP = JSAMP + 1
 310     CONTINUE
         IF (I .EQ. NCHAN) GO TO 320
         K = I + 1'
         IF (JSAMP .GE. ISTOR(K)) GO TO 840
 320     ISTOR(I) = JSAMP
 330   CONTINUE
C
CCCCC
C
C     IF ENOUGH IS DEMULT. GO TO 201 TO COPY CLOCK TO ARRAY
C     ICLK
C
CCCC
       IF (BLOCK .EQ. ITEMP) GO TO 340
CCCC
C
C
       GO TO 240
C
C
CCCC
C
C     COPY DEMULTIPLEXED CLOCK CHANNEL TO ICLK ARRAY
C
C
 340   NEND = ISTOR(JCLK)
       NSTRT = IST(JCLK)
       K = 0
       DO 350 J = NSTRT, NEND
         K = K + 1
 350   ICLK(K) = ISAM(J)
       LENGTH = K
       IF (LENGTH .GE. 30000 .OR. NBLOCK .GT. 0) GO TO 380
       WRITE (6,360)
 360   FORMAT (//'DEMULT. MORE TIME CODE TO GET>24000 SAMPLES'//)
       WRITE (6,20) TILE
C
CCCCC
C
C     RETURN TO COPY MORE CLOCK
C
CCCCC
C
C
       GO TO 240
C
CCCCC
C
C     IF ON READ OF TAPE EOF IS REACHED THEN
C     NBLOCK GIVES NO. OF BLOCKS IN THE FILE
C
CCCCC
C
 370   NBLOCK = BLOCK - 1 + NRCDS
       IF (NBLOCK .LT. ITEMP) GO TO 340
 380   IBTEMP = BLOCK - 1
       WRITE (6,390) IBTEMP, ITEMP, BLKMAX, NBLOCK
```

```
 390   FORMAT (//'BLOCKS READ IN TOTAL =', I5, /, I5,
     1        ' BLOCKS READ FOR DATA'/'BLKMAX, NBLOCK ARE ', 2I8)
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     TIME DECODING ROUTINES TRURAT AND DECODE ARE NOW CALLED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     TRIES 3 TIMES NOW TO GET SAM.RATE WITHIN A STATISTICAL ERROR OF
C     0.1%
C
 400   LERR = 0
       ISTMEM = 1
       IF (SWITS .EQ. 0) GO TO 430
       CALL FINDEX(ICLK, LENGTH, SAMRAT, INDEX)
       READ (6,420) ISDAY, ISHOUR, ISMIN
       ISSEC = 0.0
       WRITE (6,410) ISDAY, ISHOUR, ISMIN, INDEX
 410   FORMAT (/'TIME FROM FINDEX IS(DAY,HOUR,MIN)', 3I5, ' INDEX IS ',
     1        I10, /)
       GO TO 520
 420   FORMAT (I4, 2X, I2, 2X, I2)
 430   WRITE (6,440) LENGTH
 440   FORMAT (////'LENGTH OF TIME CODE INPUT TO TRURAT & DECODE IS ',
     1        I10)
       ICOUNT = 0
       ACTRAT = 0.0
       IF (SRAT .GE. 0.0) GO TO 450
       ACTRAT = -SRAT
       MEM = MEM - 1
       GO TO 470
 450   ICOUNT = ICOUNT + 1
       WRITE (6,460) ICOUNT, SRAT
 460   FORMAT (//'ATTEMPT=', I5, 1X, 'ACTRAT=', F15.6)
       ACTRAT = 0
       RESET = 1
       CALL TRURAT(RESET, ICLK, LENGTH, SRAT, ACTRAT, SD, SE, ERROR)
       IF (ERROR .NE. 0) GO TO 880
       RESET = 0
       CALL TRURAT(RESET, ICLK, LENGTH, SRAT, ACTRAT, SD, SE, ERROR)
       IF (ERROR .EQ. - 2) SRAT = ACTRAT
       IF (ICOUNT .LT. 3 .AND. ERROR .EQ. - 2) GO TO 450
       MEM = MEM + 10
       IF (ICOUNT .EQ. 3 .AND. ERROR .EQ. - 2) GO TO 470
       IF (ERROR .NE. 0) GO TO 880
C
C
C     DECODES TIME OF FIRST MINUTE AND ITS SAMPLE NUMBER(INDEX)
C
C
 470   IF (KCLK .EQ. 2) SPEC = 1
       IF (KCLK .EQ. 1) SPEC = 2
       IF (KCLK .GT. 2) GO TO 860
       WRITE (6,480)
 480   FORMAT (//'ATTEMPT 1 FOR DECODE    ')
       CALL DECODE(SPEC, ICLK, LENGTH, ACTRAT, LERR, INDEX, YEAR,
     1      MONTH, DATE, DAY, KSHOUR, KSMIN, JULDAY)
       ISDAY = DAY
       ISHOUR = KSHOUR
       ISMIN = KSMIN
       ISSEC = 0.0
       IF (LERR .NE. 0) GO TO 820
       SPEC = 0.0
       WRITE (6,490)
 490   FORMAT (//'ATTEMPT 2 FOR DECODE    ')
       CALL DECODE(SPEC, ICLK, LENGTH, ACTRAT, LERR, INDEX, YEAR,
     1      MONTH, DATE, DAY, KSHOUR, KSMIN, JULDAY)
       ISDAY = JULDAY
       ISHOUR = KSHOUR
       ISMIN = KSMIN
       ISSEC = 0.0
 500   IF (LERR .NE. 0) GO TO 820
       ISSEC = 0.0
       WRITE (6,510) ISDAY, ISHOUR, ISMIN, INDEX
 510   FORMAT (/'TIME FROM DECODE IS(DAY,HOUR,MIN)', 3I5, ' INDEX IS ',
     1        I10, /)
       WRITE (6,20) TILE
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C               END OF TIME DECODING ROUTINES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C     CALCULATE SAMPLE NUMBER OF TIME (INPUT) TO COPY TO DISC
C     CHECK ENOUGH DATA HAS BEEN COPIED ACROSS FROM TAPE SO THAT
C     NOSAM SAMPLES CAN BE COPIED TO DISC FROM THE (ISTSAM)TH SAMPLE
C
C
CCCCCCCCC
C
C   INPUT
C
CCCCCCCCC
C
 520   IF (MEM .EQ. 0 .OR. MEM .EQ. 11) GO TO 590
       IF (MEMS .EQ. 1) GO TO 590
       READ (5,570) JEVENT, ELAT, ELONG, EELEV, CLAT, CLONG, CELEV
CCCCC
```

```
C
      IF (SWIT .EQ. 1) GO TO 530
C
CCCCC
      CALL DISTAZ(CLAT, CLONG, ELAT, ELONG, CDIST, CDAZ)
C
      GO TO 550
C
 530  DO 540 J = 1, NCHAN
        CALL DISTAZ(SLAT(J), SLONG(J), ELAT, ELONG, DIST(J), DAZ(J))
 540  CONTINUE
C
 550  READ (5,580) IEDAY, IEHOUR, IEMIN, ESEC, JDUR, TCORR
      WRITE (6,560) IEDAY, IEHOUR, IEMIN, ESEC
 560  FORMAT (/'EVENT TIME IS ', I4, 2X, I2, 2X, I2, F8.4)
CCCCCCCC
C
C       COPY FROM ISEC BEFORE TIME OF SHOT
C       CORRECT FOR DRIFT IF USING AN INTERNAL CLOCK
C
CCCCCCCC
      STEMP = ESEC + TCORR - 1.0
      NOSAM = (JDUR*ACTRAT + 1)
 570  FORMAT (A4, 6F12.6)
 580  FORMAT (I4, 2X, I2, 2X, I2, F8.4, F8.2, F5.3)
CCCCC
C
C       CALCULATE FIRST SAMPLE NO. & ITS TIME
C
CCCCC
C
 590  ISTSAM = INDEX
      CALL STRSAM(ISDAY, ISHOUR, ISMIN, ISSEC, IEDAY, IEHOUR, IEMIN,
     1       STEMP, NOSAM, ISTSAM, JDUR, ACTRAT, IM, SS)
C
CCCCCC
C
C     NOW CHECK THAT HAVE NEAREST SECOND TO TIME REQUIRED
C     IF NOT TRUNCATE ICLK BY INDEX SAMPLES AND RETURN TO
C     DECODE ROUTINE : THE TRUNCATION WILL ENSURE THAT THE
C     NEXT SECOND WILL BE DECODED. MEMS IS A MEMORY CONSTANT
C     INDICATING THAT DECODE IS BEING USED AGAIN
C
CCCCC
C
      TIME = IM * 60.0 + SS
      IF (SWITS .EQ. 1) GO TO 610
      IF (ABS(TIME) .LE. 30) GO TO 610
      IF (TIME .LT. 0 .AND. MEM .EQ. 0) GO TO 1060
      IF (TIME .LT. 0 .AND. MEM .NE. 0) GO TO 610
      IF (IM .GE. 1) TIME = TIME - 60.0
      IF (IM .EQ. 0) TIME = 5.0
      TT = (TIME) * ACTRAT
      ICOUNT = INDEX + TT
      K = 0
      DO 600 J = ICOUNT, LENGTH
        K = K + 1
        ICLK(K) = ICLK(J)
 600  CONTINUE
      LENGTH = K
      IF (LENGTH .LT. (ACTRAT*130)) GO TO 610
      K = 0
      MEMS = 1
      ISTMEM = ICOUNT + ISTMEM - 1
      GO TO 470
 610  MEMS = 0
      ISTSAM = ISTMEM + ISTSAM - 1
      ISTMEM = 0
      WRITE (6,620) NOSAM, ISTSAM
 620  FORMAT (/'COPYING ', I5, ' SAMPLES FROM THE ', I10,
     1        ' TH SAMPLE FOR EACH TRACE')
C
CCCCC
C
C       DO CHECKS THAT THERE WILL BE NO PROBLEMS COPYING NOSAM SAMPLES
C       FROM THE (ISTSAM)TH SAMPLE TO DISC FOR EACH CHANNEL
C
CCCCC
      MAXSAM = (BLKMAX - 1) * 2048 / NCHAN
      IF (NOSAM .GT. MAXSAM) GO TO 960
      IOFLOW = MAXSAM - (ISTSAM + NOSAM - 1)
      IF (IOFLOW .GE. 0) GO TO 690
      NRCDS = 1 + NRCDS - IOFLOW * NCHAN / 2048
      NFILES = -1
      IF (NBLOCK .EQ. 0) GO TO 630
      NFILES = -2
      IF ((BLKMAX - 1 + NRCDS) .LE. NBLOCK) GO TO 630
      NRCDS = NBLOCK - (BLKMAX - 1)
      NOSAM = NOSAM - (NRCDS*2048/NCHAN)
      IF (NOSAM .LE. 0) GO TO 1040
 630  IF (MEM .EQ. 0 .OR. MEM .EQ. 11) GO TO 940
      IF (JFILE .EQ. 1) REWIND 4
      IF (JFILE .EQ. 1) NFILES = 0
      IF (JFILE .EQ. 1) MFILES = 0
      IF (JFILE .EQ. 1.) GO TO 640
      CALL SKIP(NFILES, 0, 4, &900)
      MFILES = 1
 640  CALL SKIP(MFILES, NRCDS, 4, &900)
      WRITE (6,650) NFILES, MFILES, NRCDS
 650  FORMAT (/'SKIPPED', I5, '+', I5, ' FILES AND', I8, ' RECORDS')
      MEM = 1
      DO 660 J = 1, 32000
 660  ISAM(J) = 0
C
C     CALCULATE ACTRAT ONLY ONCE
C
      SRAT = -ACTRAT
      LENGTH = 0
      NBLOCK = 0
      NRCDS = 0
```

```
      DO 670 J = 1, 40000
 670  ICLK(J) = 0
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     RETURN TO FIRST TAPE READ AND READ IN BLKMAX-1 BLOCKS AGAIN
C     MEM=1 : WILL NOT READ IN FROM UNIT 5 BUT WILL KEEP PREVIOUS
C             VALUES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      WRITE (6,680)
 680  FORMAT (/'MISSED EVENT RETURNING TO START AGAIN')
      GO TO 220
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
CCCCC
C
C     NOW HAS CORRECT NOSAM AND ISTIM AND ENOUGH DATA
C     COPY THESE TO DISC IN TAPEREAD FORMAT
C
CCCCC
C
 690  DO 760 J = NSTRT, NCHAN
        IBEG = IST(J) + ISTSAM - 1
        LEN = NOSAM * 2
        GAIN = 1.0
        IPOL = 1
CCCCC
C
        IF (SWIT .EQ. 1) GO TO 720
C
CCCCC
        SLAT(J) = 0
        SLONG(J) = 0
        SELEV(J) = 0
        WRITE (8,700) JEVENT, ELAT, ELONG, EELEV, IEDAY, IEHOUR,
     1  IEMIN, ESEC, JSTAT(J), SLAT(J), SLONG(J), SELEV(J), ISDAY,
     2  ISHOUR, ISMIN, ISSEC, NOSAM, ACTRAT, JDUR, GAIN, IPOL,
     3  DIST(J), DAZ(J), CDIST, CDAZ
C
 700    FORMAT (/, A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, /,
     1         A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, /, I6,
     2         F10.4, F8.2, 2X, E10.4, 2X, I4, 2F10.3, 5X, /,
     3         'DIST OF SHOT TO CENTRE OF ARRAY = ', 2F10.3)
C
        WRITE (7,710) JEVENT, ELAT, ELONG, EELEV, IEDAY, IEHOUR,
     1  IEMIN, ESEC, JSTAT(J), SLAT(J), SLONG(J), SELEV(J), ISDAY,
     2  ISHOUR, ISMIN, ISSEC, NOSAM, ACTRAT, JDUR, GAIN, IPOL,
     3  DIST(J), DAZ(J), CDIST, CDAZ
C
 710    FORMAT (A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, A4, 6X,
     1         3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, I6, F10.4, F8.2,
     2         2X, E10.4, 2X, I4, 2F10.3, 5X,
     3         'DIST OF  SHOT TO CENTRE OF ARRAY = ', 2F10.3)
C
        GO TO 750
C
 720    WRITE (8,730) JEVENT, ELAT, ELONG, EELEV, IEDAY, IEHOUR,
     1  IEMIN, ESEC, JSTAT(J), SLAT(J), SLONG(J), SELEV(J), ISDAY,
     2  ISHOUR, ISMIN, ISSEC, NOSAM, ACTRAT, JDUR, GAIN, IPOL,
     3  DIST(J), DAZ(J)
C
 730    FORMAT (A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, /, A4,
     1         6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, /, I6, F10.4,
     2         F8.2, 2X, E10.4, 2X, I4, 2F10.3)
C
        WRITE (7,740) JEVENT, ELAT, ELONG, EELEV, IEDAY, IEHOUR,
     1  IEMIN, ESEC, JSTAT(J), SLAT(J), SLONG(J), SELEV(J), ISDAY,
     2  ISHOUR, ISMIN, ISSEC, NOSAM, ACTRAT, JDUR, GAIN, IPOL,
     3  DIST(J), DAZ(J)
C
 740    FORMAT (A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, A4, 6X,
     1         3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, I6, F10.4, F8.2,
     2         2X, E10.4, 2X, I4, 2F10.3)
C
C
 750    IF (LEN .LE. 0) GO TO 920
 760  CALL WRITE(ISAM(IBEG), LEN, 0, LNUM, 7)
C
C
CCCCC
C
C     FILE AND EVENT PROCESSING COMPLETED : RETURN TO DO NEXT FILE
C
CCCCC
C
C
      MEM = 0
      WRITE (6,770) JFILE
 770  FORMAT (/'PROCESSING OF FILE', I5, ' COMPLETE'/)
      WRITE (6,780) (TILE(J),J=1,18)
 780  FORMAT (1X, 18A4, ////)
 790 CONTINUE
     REWIND 4
     GO TO 1080
C
CCCCC
C
C
C     ERROR MESSAGES
```

```
C     ------------
C
CCCCC
C
 800 WRITE (6,810)
 810 FORMAT ('***********WARNING EOF IN FIRST BLOCK***********')
     GO TO 1080
 820 WRITE (6,830) LERR, SPEC
 830 FORMAT ('ERROR IN DECODE : LERR=', I9, 'SPEC=', I2)
     GO TO 1080
 840 WRITE (6,850) NOSAM, K, ISTOR(K)
 850 FORMAT ('OVERFLOW IN ISAM FROM END OF ONE TRACE TO START OF NEXT'/
    1        'NOSAM=', I10, 'ISTOR(', I2, ')=', I10)
     GO TO 1080
 860 WRITE (6,870)
 870 FORMAT (1X, 'CHECK TIMECODE CHANNEL NUMBER ; SHOULD BE 1 OR 2 ?')
     GO TO 1080
 880 WRITE (6,890) RESET, ERROR
 890 FORMAT (1X, 'TRURAT RESET=', I2, 'ERROR=', I5)
     GO TO 1080
 900 WRITE (6,910)
 910 FORMAT (1X, 'ERROR IN TAPE SKIP')
     GO TO 790
 920 WRITE (6,930)
 930 FORMAT (1X, 'ERROR IN WRITE LEN IS NEGATIVE')
     GO TO 790
 940 WRITE (6,950)
 950 FORMAT (1X, 'TIME TO COPY FROM NEEDS CHECKING ')
     GO TO 1080
 960 WRITE (6,970)
 970 FORMAT (1X, 'CANNOT COPY OVER > 20000 SAMPLES')
     GO TO 1080
 980 WRITE (6,990)
 990 FORMAT ('EOF REACHED BEFORE SKIP COMPLETE')
     GO TO 790
1000 WRITE (6,1010)
1010 FORMAT ('BOT REACHED ON TAPE REWIND')
     GO TO 1080
1020 WRITE (6,1030)
1030 FORMAT ('UNIT NUMBER IS ASSIGNED INCORRECTLY')
     GO TO 1080
1040 WRITE (6,1050)
1050 FORMAT (/'******NOT ENOUGH DATA DIGIGTIZED*****')
     GO TO 790
1060 WRITE (6,1070)
1070 FORMAT (/'******ERROR IN RECORD SKIPS IN FILE*******')
     GO TO 790
1080 STOP
     END
C
C
C
C#######################################################
C
     SUBROUTINE STRTIM(TIME, KSDAY, KSHOUR, KSMIN, KSSEC)
```

```
C     ------------------------------------------------
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       THIS SUBROUTINES CALCULATES TIME IN DAYS,HOURS,
C       MINUTES AND SECONDS GIVEN THE TIME IN DECIMAL
C       SECONDS
C
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      REAL KSSEC
      EE = 0.0
      FF = 0.0
      GG = 0.0
      KSSEC = TIME
   10 IF (KSSEC .LT. 60.0) GO TO 20
      KSSEC = KSSEC - 60.0
      EE = 1.0 + EE
      GO TO 10
   20 KSMIN = EE
   30 IF (KSMIN .LT. 60) GO TO 40
      KSMIN = KSMIN - 60
      FF = 1.0 + FF
      GO TO 30
   40 KSHOUR = FF
   50 IF (KSHOUR .LT. 24) GO TO 60
      KSHOUR = KSHOUR - 24
      GG = 1.0 + GG
      GO TO 50
   60 KSDAY = GG
      RETURN
      END
C
C
C###############################################
C
C
      SUBROUTINE TDIFF(ID, IH, IM, SI, JD, JH, JM, SJ, T)
C     ------------------------------------------------
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      DOUBLE PRECISION SI, SJ, T, A4
      A1 = JD - ID
      A2 = JH - IH
      A3 = JM - IM
      A4 = SJ - SI
      T = ((A1*24.0 + A2)*60.0 + A3) * 60.0 + A4
      RETURN
      END
C
C
C###############################################
C
```

```
C
      SUBROUTINE STRSAM(ISDAY, ISHOUR, ISMIN, ISSEC, JSDAY, JSHOUR,
     1            JSMIN, JSSEC, JSAMP, ISTSAM, JDUR, ACTRAT, IM, SS)
C     ___________________
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
      INTEGER SRTSAM
      REAL JSSEC, JDUR, ISSEC, KSSEC
C
C
C      CALCULATE FIRST SAMPLE NUMBER IN SAM TO COPY OVER FROM
C      AND CALCULATE TIME OF FIRST SAMPLE
C
C
   10 CALL TDIFF(ISDAY, ISHOUR, ISMIN, ISSEC, JSDAY, JSHOUR, JSMIN,
     1     JSSEC, T)
      IF (T .LT. 0) GO TO 20
C
      TT = ACTRAT * T
      NSAM = TT + 1
      ISTSAM = NSAM + ISTSAM - 1
      TIME = (NSAM - 1) / ACTRAT
      CALL STRTIM(TIME, ID, IH, IM, SS)
      ISDAY = ID + ISDAY
      ISHOUR = IH + ISHOUR
      ISMIN = IM + ISMIN
      ISSEC = SS + ISSEC
      GO TO 30
C
   20 T = -T
      TT = ACTRAT * T
      NSAM = TT + 1 + 1
      ISTSAM = -NSAM + ISTSAM + 1
      TIME = (NSAM - 1) / ACTRAT
      CALL STRTIM(TIME, ID, IH, IM, SS)
      ISDAY = -ID + ISDAY
      ISHOUR = -IH + ISHOUR
      ISMIN = -IM + ISMIN - 1
      ISSEC = 60.0 - SS
C
   30 WRITE (6,40) ISTSAM, IM, SS
   40 FORMAT (1X, 'START SAMPLE IS', I8, ' WHICH IS', I2, ' MIN ', F8.4,
     1       ' SECS  FROM NEAREST MIN')
C
      RETURN
      END
C
C
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
C
```

```
      SUBROUTINE DISTAZ(AL, OL, BL, CL, DIST, DAZ)
C     ____________________________________________
C
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     CALCULATES DISTANCE BETWEEN SHOT AND STATION FROM DECIMAL DEGREES
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     ORIGINAL VERSION G.K.WESTBROOK
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      DISTANCE CALCULATOR ACCURATE TO 1 IN 10**7 AT 1600 KM
C     USES FORMULA OF ROBBINS, A.R. 'LONG LINES ON THE
C     SPHEROID' EMP. SVY. REV. 125 1962. WRITTEN FOR
C     FINDING DISTANCES BETWEEN RECEIVING STATIONS AND SHOT
C     POINTS IN SEISMIC EXPERIMENTS, BUT CAN BE USED FOR ANY
C     DISTANCE CALCULATION. NLP = NO. OF STATIONS.  NSP = NO
C     OF SHOTS.  NL(I) = IDENTIFICATION NO. OF ITH RECEIVING
C     STATION.  AL= DEGREES OF LATITUDE.  AM = MINUTES OF LA
C     TITUDE.  OL = DEGREES OF LONGITUDE.  OM = MINUTES OF L
C     TITUDE.  OL = DEGREES OF LONGITUDE.  OM = MINUTES OF
C     LONGITUDE.  NS(I) = IDENTIFICATION NO. OF ITH SHOT.
C     LONGITUDE : EAST IS POSITIVE
C     LATITUDE : NORTH IS POSITIVE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DOUBLE PRECISION DL, HL, ERQ, PRQ, A, B, SS, VS, SI, V, CO, S,
     1       SDL, CDL, DIF, S1, C1, G, GA, EC, EL, TA, GS, FL, GL
      DIMENSION NL(50), A(2,50), NS(500), B(2,500), SI(50), CO(50),
     1          V(50), VS(500), SS(500), GA(50), TA(500), GS(500),
     2          TITLE(20), NT(500)
      DTR = 0.01745329
      ER = 6378.160
      PR = 6356.775
      ERQ = ER ** 2
      PRQ = PR ** 2
      EC = (ERQ - PRQ) / ERQ
      EL = (ERQ - PRQ) / PRQ
      DL = AL * DTR
      HL = OL * DTR
      SS(1) = DSIN(DL)
      TA(1) = SS(1) / DCOS(DL)
      VS(1) = ER / DSQRT(1 - EC*SS(1)**2)
      GS(1) = DSQRT(EL*SS(1)**2)
      FL = BL * DTR
      GL = CL * DTR
      B(1,2) = FL
      B(2,2) = GL
      S1 = DSIN(FL)
      C1 = DCOS(FL)
      V1 = ER / DSQRT(1 - EC*S1**2)
      DIF = HL - GL
      CDL = DCOS(DIF)
      SDL = DSIN(DIF)
```

```
      SS(2) = S1
      VS(2) = V1
      TA(2) = S1 / C1
      GS(2) = DSQRT(EL*S1**2)
      G = GS(2)
      T = (1.0 - EC + EC*V1*S1/(VS(1)*SS(1))) * TA(1)
      AT = ATAN(T)
      PAZ = C1 * T - S1 * CDL
      SL = SDL
      AZ = ATAN2(SL,PAZ)
      HQ = EL * C1 ** 2 * COS(AZ) ** 2
      H = SQRT(HQ)
      BAZ = ABS(AZ) - 3.14159
      IF (BAZ .GT. - 1.0 .OR. BAZ .LT. - 2.14159) GO TO 10
      FD = SDL * COS(AT) / SIN(AZ)
      D = ARSIN(FD)
      GO TO 20
   10 D = ARSIN(COS(AT)*PAZ/COS(AZ))
   20 DIST = V1 * D * (1.0 - D**2*HQ*(1.0 - HQ)/6.0 + D**3*G*H*(1.0 - 2.
     10*HQ)/8.0 + D**4*(HQ*(4.0 - 7.0*HQ) - 3.0*G**2*(1.0 - 7.0*HQ))/
     2120.0 - D**5*G*H/48.0)
      DAZ = AZ / DTR
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FINDEX(ICLK, LENGTH, SAMRAT, INDEX)
C       ______________________________________________
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      INTEGER LENGTH
      INTEGER*2 ICLK
      DIMENSION ICLK(LENGTH), EPOCH(240), EPLEN(240)
C
      ISUM = 0
      DO 10 I = 1, LENGTH
        ISUM = ICLK(I) + ISUM
   10 CONTINUE
      IDISC = ISUM / LENGTH
      K = 0
      DO 20 I = 1, LENGTH
        J = I - 1
        IF (ICLK(I) .LE. IDISC .AND. ICLK(J) .GT. IDISC) EPLEN(K) = J -
     1  EPOCH(K)
        IF (ICLK(I) .LE. IDISC) GO TO 20
        IF (ICLK(J) .GT. IDISC) GO TO 20
        K = K + 1
        EPOCH(K) = I
   20 CONTINUE
      NSEC = K
      K = 0
      DO 30 I = 1, NSEC
        K = K + 1
        J = I - 1
        LEN = EPLEN(J) + EPLEN(I)
        SECLEN = LEN / SAMRAT
        IDIFF = EPOCH(I) - EPOCH(J) + IDIFF
        IF (SECLEN .GT. 0.7 .AND. SECLEN .LT. 0.9) GO TO 40
   30 CONTINUE
   40 INDEX = EPOCH(J)
      SRAT = IDIFF / K
      WRITE (6,50) SRAT
   50 FORMAT (/'APPROX SAMRAT FROM FINDEX IS ', F10.4)
      RETURN
      END
      SUBROUTINE DECODE(SPEC, ARRAY, LENGTH, SAMRAT, ERROR, INDEX, YEAR,
     1           MONTH, DATE, DAY, HOUR, MINUTE, JULDAY)
C
C
C TIME CODE DECODING ROUTINE VERSION 1.3     1/9/83
C
C THIS SUBROUTINE ENABLES PULSE WIDTH ENCODED TIME CODES
C IN DIGITAL FORM TO BE DECODED.
C
C TIME CODES ETC ARE SPECIFIED IN DATA ARRAYS SET UP WITHIN
C THE SUBROUTINE AND A SWITCH IS USED TO SELECT THE APPROPRIATE
C SPECIFICATION.
C
C ARRAY CONTAINS I*2 DATA
C LENGTH IS LENGTH OF ARRAY
C SAMRAT IS R*4 SAMPLING RATE
C ERROR IS ERROR CODE RETURNED
C
C 1=MORE DATA REQUESTED
C 0=NORMAL RETURN
C -VE= ERROR IN DECODING
C
C
C SPEC = SWITCH TO SELECT SPEC, 0=> USE PREVIOUSLY SET SPEC AND SAMRAT
C                               1=> USE SPEC 1 ETC.
C
C
C
C NOTE 1
C  PWSPEC CONTAINS PULSE WIDTH SPECS IN SECS, WIDTH FOLLOWED BY
C TOLERANCE.
C PWSPEC(X,Y) ACCESSES ELEMENT X OF SPEC Y
C
C NOTE 2
C INTEGER PULSE WIDTHS MAXIMUM,MINIMUM OF PULSE N OF SPEC
C IPW(MM,N)
C
C VERSION 1.1
C MODIFIED TO TAKE INTO ACCOUNT ROUNDING ERRORS ALSO WIDER TOLERANCES
C ONLY SUPPORTS VELA AND MSF (BOTH TYPES OF MIN PULSE FOR MSF)
```

```
C
C VERSIONN 1.2
C MINIMUM RANGE ON DISCRIMINATION LEVEL  APPLIED TO OVERCOME
C PROBLEMS WITH NO SIGNAL ON MSF CHANNELS ETC.
C
C DECLARATIONS
      IMPLICIT INTEGER*2(A - Z)
      INTEGER*4 INDEX, JOFSET, JONE, JWC, JTEMP, LENGTH
      REAL SAMRAT, SUM, PWSPEC, PWSPC1, PWSPC2, PWSPC3, PWSPC4, SUMH,
     1     SUML
      DIMENSION ARRAY(LENGTH)
C
C
      DIMENSION VAR(10), LASDAY(12)
      DIMENSION PWSPEC(12,4)
      DIMENSION IPW(2,8)
      DIMENSION VASPEC(60,4)
      DIMENSION PNUM(4)
      DIMENSION VASPC1(60), VASPC2(60), VASPC3(60), VASPC4(60)
      DIMENSION PWSPC1(12), PWSPC2(12), PWSPC3(12), PWSPC4(12)
      DIMENSION MUSPEC(60,4)
      DIMENSION MUSPC1(60), MUSPC2(60), MUSPC3(60), MUSPC4(60)
C
C
      EQUIVALENCE (MUSPC1(1),MUSPEC(1,1)), (MUSPC2(1),MUSPEC(1,2)),
     1            (MUSPC3(1),MUSPEC(1,3)), (MUSPC4(1),MUSPEC(1,4))
      EQUIVALENCE (PWSPC1(1),PWSPEC(1,1)), (PWSPC2(1),PWSPEC(1,2)),
     1            (PWSPC3(1),PWSPEC(1,3)), (PWSPC4(1),PWSPEC(1,4))
      EQUIVALENCE (VASPC1(1),VASPEC(1,1)), (VASPC2(1),VASPEC(1,2)),
     1            (VASPC3(1),VASPEC(1,3)), (VASPC4(1),VASPEC(1,4))
C
      DATA LASDAY /31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/
C
      WRITE (6,80)
C
C SPECS FOR MSF CODE
C
      PNUM(1) = 58
      DATA PWSPC1 /0.050, 1.100, 0.100, 0.040, 0.200, 0.040, 0.300,
     1     0.040, 4*0.000/
      DATA VASPC1 /16*0, 8*1, 5*2, 6*3, 3*4, 6*5, 7*6, 6*0, 7, 0, 0/
      DATA MUSPC1 /16*0, 80, 40, 20, 10, 8, 4, 2, 1, 10, 8, 4, 2, 1, 20,
     1     10, 8, 4, 2, 1, 4, 2, 1, 20, 10, 8, 4, 2, 1, 40, 20, 10, 8,
     2     4, 2, 1, 6*0, 1, 0, 0/
C
C SPECS FOR VELA CODE (GEOSTORE INTERNAL CLOCK)
C
      PNUM(2) = 56
      DATA PWSPC2 /0.100, 1.100, 0.200, 0.060, 0.500, 0.150, 0.800,
     1     0.100, 4*0.000/
      DATA VASPC2 /4*1, 2*0, 2*2, 2*0, 8*2, 4*0, 6*3, 3*0, 7*4, 2*0,
     1     8*5, 2*0, 4*5, 6*0/
      DATA MUSPC2 /8, 4, 2, 1, 0, 0, 200, 100, 0, 0, 80, 40, 20, 10, 8,
     1     4, 2, 1, 4*0, 20, 10, 8, 4, 2, 1, 3*0, 40, 20, 10, 8, 4, 2,
     2     1, 0, 0, 800, 400, 200, 100, 80, 40, 20, 10, 0, 0, 8, 4, 2,
     3     1, 6*0/ '
C
C*************************************************
C RESET OR DECODE ——ENTRY POINT——
C*************************************************
C
      ERROR = 0
      WRITE (6,10) SPEC, LENGTH, SAMRAT
   10 FORMAT (1X, 'SPEC,LENGTH,SAMRAT ARE ', 2I10, F15.6)
      IF (LENGTH .NE. 24000) GO TO 20
C
C      DEBUG TOOL TO LIST TIME CODE
C
C      NEND=0
C      DO 77 I=1,50
C        NST=NEND+1
C        NEND=NST+9
C        WRITE(6,29) (ARRAY(J),J=NST,NEND)
C29      FORMAT(1X,10I7)
C77    CONTINUE
C
C
   20 JONE = 1
      IF (SPEC) 100, 100, 30
C
C FIX DISCRIMINATOR LEVEL AND RESET IPW PARAMETERS AND ALL REGISTERS
C
   30 SPSET = SPEC
      DO 40 I = 1, 10
   40 VAR(I) = 0
      SUM = 0.00
      SUMH = 0.00
      SUML = 0.00
      DO 50 N = 1, LENGTH
   50 SUM = SUM + ARRAY(N)
      AVE = INT(SUM/LENGTH)
      DO 60 N = 1, LENGTH
        IF (ARRAY(N) .GE. AVE) SUMH = (SUMH + ARRAY(N)) / 2.00
   60 IF (ARRAY(N) .LT. AVE) SUML = (SUML + ARRAY(N)) / 2.00
      SPREAD = INT(SUMH - SUML)
      DISC = AVE - SPREAD / 10
      IF (SPREAD .LT. 256) ERROR = -2
      WRITE (6,70) SPREAD, ERROR
   70 FORMAT (1X, 'SPREAD & ERROR = ', 2I10)
      IF (ERROR .NE. 0) RETURN
C
C NOW THE IPW ARRAY
   80 FORMAT ('CHECK')
      WRITE (6,80)
C
      IPW(1,1) = INT(SAMRAT*PWSPEC(1,SPEC) + 0.5)
      IPW(2,1) = IPW(1,1)
      IPW(1,2) = INT(SAMRAT*PWSPEC(2,SPEC) + 0.5)
      IPW(2,2) = IPW(1,2)
      L = 3
```

```
      DO 90 I = 3, 11, 2
        IPW(1,L) = INT(SAMRAT*(PWSPEC(I,SPEC) - PWSPEC(I + 1,SPEC)) + 0.
     1  5)
        IPW(2,L) = INT(SAMRAT*(PWSPEC(I,SPEC) + PWSPEC(I + 1,SPEC)) + 0.
     1  5)
   90 L = L + 1
C        WRITE(6,885)(IPW(1,KL),KL=1,5)
C        WRITE(6,886)(IPW(2,KL),KL=1,5)
C885     FORMAT(1X,'DEBUG MODE  MIN PULSES',5I5)
C886     FORMAT(1X,'DEBUG MODE  MAX PULSES',5I5)
      WRITE (6,80)
C
      JOFSET = 1
      MINFND = 0
      SECGAT = 0
      PCOUNT = 0
      WCOUNT = 0
      RUNWAY = INT(2.1*SAMRAT)
      SAMPLE = 1
      ERROR = 0
      RETURN
C
C MAIN DECODING SECTION                    1
C
  100 IF (SAMPLE .LE. LENGTH) GO TO 110
      SAMPLE = 1
      ERROR = 1
      RETURN
C
  110 IF (ARRAY(SAMPLE) .LT. DISC) SAM = 1
      IF (ARRAY(SAMPLE) .GE. DISC) SAM = 0
      IF (MINFND .NE. 0) GO TO 120
      GO TO (390, 470, 540, 550), SPSET
C
C MINUTE /INDEX PULSE ALREADY FOUND
C
  120 IF (WCOUNT .GT. 0) GO TO 160
C
C COUNTER NOT RUNNING
C
  130 IF (SAM) 140, 140, 150
  140 IF (RUNWAY .LE. 0 .AND. MINFND .EQ. 1) GO TO 240
      GO TO 250

  150 IF (SECGAT .LE. 0) WCOUNT = WCOUNT + 1
      GO TO 250
C
C COUNTER RUNNING
C
  160 IF (SAM) 170, 170, 200
  170 IF (WCOUNT .GT. IPW(1,1)) GO TO 190
C
  180 LAHEAD = SAMPLE + IPW(1,1) - 1
      IF (LAHEAD .GT. LENGTH) LAHEAD = SAMPLE
      IF (ARRAY(LAHEAD) .LT. DISC) GO TO 200
      GO TO 140
C
C COUNT>MINIM PULSE
C
  190 LAHEAD = SAMPLE + IPW(1,1) - 1
      IF (LAHEAD .GT. LENGTH) LAHEAD = LENGTH
      IF (ARRAY(LAHEAD) .LT. DISC) GO TO 200
      GO TO 210
C
C COUNT SAMPLE
C
  200 WCOUNT = WCOUNT + 1
      GO TO 250
C
C PULSE WIDTH DECODING
C
  210 DO 220 N = 3, 7
        IF (WCOUNT .GE. IPW(1,N) .AND. WCOUNT .LE. IPW(2,N))
     1     GO TO 230
  220 CONTINUE
      ERROR = -1
      WCOUNT = 0
      PCOUNT = 0
      SECGAT = 0
      MINFND = 0
      RUNWAY = 0
      SAMPLE = 1
      JOFSET = 1
      RETURN
C
  230 PULSE = N - 3
      VARNUM = VASPEC(PCOUNT,SPSET)
      IF (VARNUM .EQ. 0) GO TO 240
      VAR(VARNUM) = VAR(VARNUM) + PULSE * MUSPEC(PCOUNT,SPSET)
C        WRITE(6,887)PCOUNT,WCOUNT,PULSE,MUSPEC(PCOUNT,SPSET),
C     1  VASPEC(PCOUNT,SPSET)
C887     FORMAT(1X,'MODEBUG  PULSE,W,ID,MUL,VA',5I5)
  240 PCOUNT = PCOUNT + 1
      IF (PCOUNT .GT. PNUM(SPSET)) GO TO 270
      RUNWAY = IPW(1,2)
      SECGAT = INT(SAMRAT) - IPW(1,1) - WCOUNT
      WCOUNT = 0
C
C NOW INCREMENT SAMPLE AND AROUND WE GO
C
  250 IF ((SAMPLE + SECGAT .LE. LENGTH) .AND. SECGAT .GT. 1)
     1   GO TO 260
      SAMPLE = SAMPLE + 1
      RUNWAY = RUNWAY - 1
      SECGAT = SECGAT - 1
      IF (MINFND .EQ. 0) JOFSET = JOFSET + JONE
      GO TO 100
C
C SPEED UP READING
C
```

```
 260 SAMPLE = SAMPLE + SECGAT
     RUNWAY = RUNWAY - SECGAT
     JTEMP = SECGAT
     IF (MINFND .EQ. 0) JOFSET = JOFSET + JTEMP
     SECGAT = 0
     GO TO 100
C
C ALL DECODED NOW TIDY UP AND STANDARDISE TIME FORMATS
C
 270 MINFND = 0
     PCOUNT = 0
     WCOUNT = 0
     RUNWAY = 0
     SECGAT = 0
     SAMPLE = 1
     JOFSET = 1
C
C
C*******************************************************
C SPEC DEPENDENT CODE FOR STANDARDISATION OF TIME CODES
C*******************************************************
C
C
C
     GO TO (280, 340, 370, 380), SPSET
C
C CODE FOR MSF CLOCK
C GMT TIME RETURNED
C
 280 VAR(6) = VAR(6) - 1
     IF (VAR(6) .GE. 0) GO TO 310
     VAR(6) = 0
     VAR(5) = VAR(5) - 1
 290 IF (VAR(5) .GE. 0) GO TO 310
     VAR(5) = 0
     VAR(4) = VAR(4) - 1
     IF (VAR(4) .LT. 0) VAR(4) = 7
     VAR(3) = VAR(3) - 1
     IF (VAR(3) .GT. 0) GO TO 310
     VAR(2) = VAR(2) - 1
     IF (VAR(2) .GT. 0) GO TO 300
     VAR(2) = 1
     VAR(1) = VAR(1) - 1
 300 VAR(3) = LASDAY(VAR(2))
 310 IF (VAR(7) .EQ. 0) GO TO 320
     VAR(7) = 0
     VAR(5) = VAR(5) - 1
     GO TO 290
 320 YEAR = VAR(1) + 1900
     MONTH = VAR(2)
     DATE = VAR(3)
     DAY = VAR(4)
     HOUR = VAR(5)
     MINUTE = VAR(6)
     JULDAY = 0
     VARIC = VAR(2) - 1
     DO 330 J = 1, VARIC
 330 JULDAY = JULDAY + LASDAY(J)
     JULDAY = JULDAY + DATE
     ERROR = 0
     RETURN
C
C
C
 340 YEAR = VAR(1)
     JULDAY = VAR(2)
     TEMP = 0
     DO 350 J = 1, 12
       TEMP = TEMP + LASDAY(J)
       IF (TEMP .GE. VAR(2)) GO TO 360
 350 CONTINUE
C
 360 MONTH = J
     DATE = TEMP - VAR(2) + 1
     DAY = -1
     HOUR = VAR(3)
     MINUTE = VAR(4)
     ERROR = 0
C
     RETURN
C
C
 370 CONTINUE
 380 ERROR = -2
C
     RETURN
C
C
C
C
C
C ***************************************************************
C ***
C INDEX PULSE LOCATING ROUTINES ,DEPENDENT ON SPEC
C ***************************************************************
C ***
C
C CODE FOR MSF MINUTE PULSE WITH 100HZ BURST AS INDEX
 390 MINMAX = INT(0.530*SAMRAT + 0.5)
     MINMIN = INT(0.470*SAMRAT + 0.5)
     DTIME = INT(0.080*SAMRAT)
     IF (SAM) 410, 410, 400
 400 WCOUNT = WCOUNT + 1
     DEDCNT = DTIME
     PS = 1
     GO TO 250
 410 IF (WCOUNT) 420, 420, 430
 420 PS = 0
     GO TO 250
 430 IF (WCOUNT .GE. MINMIN .AND. WCOUNT .LE. MINMAX) GO TO 460
```

```
      IF (DEDCNT) 440, 440, 450
  440 WCOUNT = 0
      MINFND = 0
      GO TO 250
  450 WCOUNT = WCOUNT + 1
      DEDCNT = DEDCNT - 1
      GO TO 250
C
C MINUTE FOUND
C
  460 JXC = WCOUNT
      INDEX = JOFSET - JWC
      WCOUNT = 0
      PCOUNT = 1
      MINFND = 1
      RUNWAY = IPW(1,2)
      GO TO 250
C
C CODES FOR OTHER TIME CODES
C VELA
C
  470 IF (WCOUNT .GT. 0) GO TO 490
  480 IF (SAM .EQ. 1) WCOUNT = WCOUNT + 1
      GO TO 250
  490 IF (SAM .EQ. 0) GO TO 500
      WCOUNT = WCOUNT + 1
      GO TO 250
  500 LAHEAD = SAMPLE + IPW(1,1)
      IF (LAHEAD .GT. LENGTH) LAHEAD = SAMPLE
      IF (ARRAY(LAHEAD) .LT. DISC) GO TO 480
  510 FORMAT (' DEBUG MODE  PULSE WIDTH,PP REF PULSE', 2I7)
      IF (WCOUNT .GE. IPW(1,5) .AND. WCOUNT .LE. IPW(2,5))
     1   GO TO 520
      WCOUNT = 0
      PP = 0
      GO TO 250
  520 IF (PP .EQ. 1) GO TO 530
      WCOUNT = 0
      PP = 1
      GO TO 250
  530 JWC = WCOUNT
      INDEX = JOFSET - JWC
      WCOUNT = 0
      PCOUNT = 1
CD       PAUSE 'MINFND'
      MINFND = 1
      RUNWAY = IPW(1,2)
      PP = 0
      GO TO 250
C
  540 ERROR = -2
  550 ERROR = -2
      RETURN
      END
      SUBROUTINE TRURAT(RESET, ARRAY, LENGTH, SRAT, ACTRAT, SD, SE,
     1            ERROR)
C
C ROUTINE TO DETERMINE TRUE SAMPLING RATE ACTRAT FROM TIME CODE
C RETURNS THE STANDARD DEVIATION AND ERROR IN HZ
C
C VERSION 1.3 1/10/83
C
C -VE GOING PULSES OF TIME CODE DATA
C
C
C RESET =0 CONTROL CODE NORMALLY ZERO
C  RESET =1 => RESET ROUTINE WHEN MORE DATA REQUESTED OR CALCULATE
C DISCRIM
C       LEVELS
C
C ERROR =1 => NOT ENOUGH CODE,ROUTINE MAY BE RE-ENTERED
C       WITH ANOTHER BUFFER FULL OF CODE.
C ERROR=-1 =>SAMRAT TOO HIGH
C ERROR=-3 => RANGE TOO LOW
C ERROR=-2 =>SAMRAT=SRAT IE ESTIMATED
C ERROR=-4 => MISSING PULSES
C
      IMPLICIT INTEGER*2(A - Z)
      INTEGER*4 LENGTH, DIF
      REAL SAMRAT, SUM, SUMH, SUML, ACTRAT, SD, SE, AMEAN, RTEMP
      REAL SRAT, ABS, DELTA, EPOCH, BCOUNT, XLEN, DELT, TNUOC, R1, R2
      REAL*8 SMEAN, NMEAN, DBLE
      DIMENSION ARRAY(LENGTH), EPOCH(240)
C
C
C*************************************************
C  ---ENTRY POINT---
C*************************************************
C CALL COUNT
      ICC = 0
      ICC = ICC + 1
      ICCC = 1
      IF (RESET .EQ. 0) ICCC = 2
      WRITE (6,10) ICCC
   10 FORMAT (1X, 'TRURAT', I7)

C
C FIX DISCRIMINATOR LEVEL AND RESET IPW PARAMETERS AND ALL REGISTERS
C
      SAMRAT = SRAT
      ERROR = 0
      IF (RESET .EQ. 0) GO TO 120

   20 SUM = 0.00
      SUMH = 0.00
      SUML = 0.00
      NAL = 0
      NBL = 0
      IF (SAMRAT .LE. 1000.00) GO TO 30
      ERROR = -1
```

```
      RETURN
C
C DISCRIMINATOR CALCULATION
C
   30 DO 40 N = 1, LENGTH
   40 SUM = SUM + ARRAY(N)
      AVE = INT(SUM/LENGTH)

      DO 60 N = 1, LENGTH
        IF (ARRAY(N) .LT. AVE) GO TO 50
        SUMH = SUMH + ARRAY(N)
        NBL = NBL + 1
        GO TO 60

   50   SUML = SUML + ARRAY(N)
        NAL = NAL + 1
   60 CONTINUE
C
C CHECK THAT WE HAVE SUFFICIENT BASELINE IE AT LEAST HALF SAMPLES
C
      IF (NAL .GT. 0 .AND. NBL .GE. LENGTH/2) GO TO 70
      ERROR = -3
      RETURN


   70 SUMH = SUMH / NBL
      SUML = SUML / NAL
      DIFF = INT(SUMH - SUML)
C
C NOW THE VARIENCE
C
      SUML = 0.0
      DO 80 N = 1, LENGTH
        IF (ARRAY(N) .LT. AVE) GO TO 80
        SUML = SUML + (SUMH - ARRAY(N)) ** 2
   80 CONTINUE
      SPREAD = SQRT(SUML/(NBL - 1))
      DISC = SUMH - SPREAD

      WRITE (6,90) AVE, SUMH, SPREAD, DISC, DIFF
   90 FORMAT (' Average ', I7, ' High base =', F10.3, ' Spread', I7,
     1        ' Disc ', I7, 'Diff', I6)

C
C CHECK FOR SIGNAL PRESENT
C
      DIF = DIFF
      IF (IABS(DIF) .LT. 128 .OR. SPREAD  GT. IABS(DIF)) ERROR = -3
      IF (ERROR .LT  0) RETURN

C
C ZERO COUNTERS
C
  100 POCUNT = 1
      WCOUNT = 0
      MINPUL = INT(SAMRAT*0.05 + 0.5)
      RUNWAY = INT(SAMRAT*5.00)
      SAMPLE = 1
      BCOUNT = 0.0
      IARM = 0
      XLEN = SAMRAT * 300.0
      SECGAT = 0
      ERROR = 0
      DO 110 K = 1, 240
  110 EPOCH(K) = 0.00
      RETURN
C
C ALTERNATE- NORMAL ENTRY
C
C NOW START COUNTING
C
  120 IF (SAMPLE .LE. LENGTH) GO TO 140
      SAMPLE = 1
      BCOUNT = BCOUNT + LENGTH

C CHECK THAT WE HAVE NOT MISSED PULSES
      IF (BCOUNT .GT. XLEN) ERROR = -4
      IF (ERROR .LT. 0) RETURN
      WRITE (6,130) POCUNT
  130 FORMAT (' POCUNT ', I7)
      ERROR = 1
      RUNNIN = 1
      RETURN

  140 IF (SECGAT) 150, 150, 240
  150 IF (RUNWAY) 250, 250, 160
  160 IF (ARRAY(SAMPLE) .GT. DISC) GO TO 170

C
C PULSE
C
      IF (IARM .EQ. 0) GO TO 240
      WCOUNT = WCOUNT + 1
      IF (WCOUNT .GT. 1) GO TO 240
      IF (PSAMP .LE. DISC) GO TO 240

      R1 = (DISC - PSAMP)
      R2 = (ARRAY(SAMPLE) - PSAMP)
      DELT = R1 / R2
      DELTA = ABS(DELT)
      EPOCH(POCUNT) = BCOUNT + SAMPLE - 1.0 + DELTA
C       WRITE(6,1005) EPOCH(POCUNT),DELTA,PSAMP,ARRAY(SAMPLE)
C1005  FORMAT(' EPOCH AT',F14.3,'  DELTA',F6.3,' PSAMP',I7,' SAMP',I7)
      GO TO 240
C
C BASE LINE
C
  170 IARM = 1
      IF (WCOUNT) 240, 240, 180
```

```
C
C END OF PULSE
C
 180 LAHEAD = INT(SAMRAT*0.025) + SAMPLE
     IF (LAHEAD .GT. LENGTH) LAHEAD = LENGTH

     DO 190 K = SAMPLE, LAHEAD
       IF (ARRAY(LAHEAD) .LE. DISC) GO TO 200
 190 CONTINUE
     GO TO 210
C
C FALSE ALARM
C
 200 WCOUNT = WCOUNT + 1
     GO TO 240
C
C REAL PULSE, FILTER SPIKES
C
 210 IF (WCOUNT .GE. MINPUL) GO TO 220
     WCOUNT = 0
     GO TO 240

C
C RESET FOR NEXT PULSE
C
 220 SECGAT = INT(SAMRAT*0.80) - WCOUNT

     WCOUNT = 0
     RUNWAY = INT(SAMRAT*5.0)

 230 PCOUNT = PCOUNT + 1
     IF (PCOUNT .GT. 240) GO TO 270
     RUNNIN = 0

 240 PSAMP = ARRAY(SAMPLE)

     SAMPLE = SAMPLE + 1
     SECGAT = SECGAT - 1
     RUNWAY = RUNWAY - 1
     GO TO 120

C MISSING PULSE - FLAG THE ERROR
 250 IF (PCOUNT .GT. 2) GO TO 260
     SECGAT = 0
     RUNWAY = INT(SAMRAT*2.5)
     GO TO 240

 260 ERROR = -4
     IARM = 0
     RUNNIN = 0
     RETURN

C NOW DO THE STATS TO CALCULATE AVERAGE,SD&SE
 270 AMEAN = 0.000
     SMEAN = 0.000D00
     COUNT = 0

     DO 280 K = 2, 240
       EPOCH(K - 1) = EPOCH(K) - EPOCH(K - 1)
       IF (EPOCH(K - 1) .GT. (SRAT*11.0/10.0) .OR. EPOCH(K - 1) .LT. (
    1     SRAT*9.0/10.0)) GO TO 280
       COUNT = COUNT + 1
       SMEAN = SMEAN + DBLE(EPOCH(K - 1))
 280 CONTINUE

     AMEAN = SMEAN / COUNT

     WRITE (6,290) AMEAN, SMEAN, COUNT
 290 FORMAT (' MEAN SAMRAT Before STATS', F14.6, ' SMEAN   ', E14.4,
    1        I10)
     SMEAN = 0.000D00
     COUNT = 0

     DO 300 K = 1, 239
       IF (EPOCH(K - 1) .GT. (SRAT*11.0/10.0) .OR. EPOCH(K - 1) .LT. (
    1     SRAT*9.0/10.0)) GO TO 300
       COUNT = COUNT + 1
       RTEMP = (AMEAN - EPOCH(K)) ** 2
 300 SMEAN = SMEAN + (DBLE(RTEMP))
     RTEMP = COUNT - 1
     SD = (SMEAN/DBLE(RTEMP)) ** 0.5

C NOW RESCAN DATA REMOVING WILD POINTS
     COUNT = 0
     NMEAN = 0.0

     DO 310 K = 1, 239
       IF (EPOCH(K) .GT. (AMEAN + SD*0.50) .OR. EPOCH(K) .LT. (AMEAN -
    1     SD*0.50)) GO TO 310
       COUNT = COUNT + 1
       NMEAN = NMEAN + DBLE(EPOCH(K))
 310 CONTINUE
     ACTRAT = NMEAN / COUNT
     IF (COUNT .LT. 200) ERROR = -2
     WRITE (6,320) COUNT
 320 FORMAT (' COUNT ', I7)
     WRITE (6,330) ACTRAT
 330 FORMAT (' MEAN SAMRAT AFTER STATS', F14.6)
     NMEAN = 0.0
     COUNT = 0

C
C DO STATS FOR STANDARD ERROR
C
     DO 340 K = 1, 239
       IF (EPOCH(K) .GT. (AMEAN + SD*0.50) .OR. EPOCH(K) .LT. (AMEAN -
    1     SD*0.50)) GO TO 340
       COUNT = COUNT + 1
       RTEMP = (ACTRAT - EPOCH(K)) ** 2
       NMEAN = NMEAN + (DBLE(RTEMP))
```

```
 340 CONTINUE

     RTEMP = COUNT - 1
     SD = (NMEAN/DBLE(RTEMP)) ** 0.5
     TNUOC = COUNT
     SE = SD / (SQRT(TNUOC))
     RUNNIN = 0
     WRITE (6,350) SE, SD
 350 FORMAT (' STANDARD ERR', F14.8, ' SD ', F14.8)

C ALLOW ONLY +- 0.1% ERROR IN DETERMINATION
     IF (100.*SE/ACTRAT .GT. 0.1) GO TO 370

C 10% TOLERANCE ON INPUT SAMRAT
     IF (ACTRAT .LT. (SRAT - SRAT/10.) .OR. ACTRAT .GT. (SRAT + SRAT/
    1   10.)) GO TO 360

C       PAUSE 'RATE WITHIN LIMITS'
     RETURN

C  ESTIMATED SAMRAT OR OUTSIDE ACCEPTABLE LIMITS OF ACCURACY BY
C SUBROUTINE
 360 ACTRAT = SRAT
 370 ERROR = -2
     RETURN
     END
C
```

```
C
CA    AHJ Lewis
C
C           VERSION ON 14TH MAY 1985
C*********************************************
C
C           CDIST
C           ------
C             BY
C
C       A.H.J.L. LEWIS
C
CCCCCCCC
C
C    PROGRAMME TO CONVERT SHOT STATION DISTANCES FOR
C    COMMON STATION SECTION TO OMP DISTANCES FROM M25B.
C
C    UNIT 1 = GPT9:CDS - FILE THAT CONTAINS TRACES
C    UNIT 4 = GPT9:XMAT-M25B - FILE THAT CONTAINS
C                              DISTANCES TO M25B.
C    UNIT 7 = OUTPUT OF DETAILS AND SUMMARY OF
C             DISTANCES CHANGEDSD.
C
C*********************************************
C
C
     DIMENSION DIST25(160)
     REAL*4 KSSEC
     INTEGER*4 A(160)
C
C   SKIP HEADINGS
C
     WRITE(6,1)
 1   FORMAT('ENSURE DATA FILE HAS SEQUENTIAL LINE NUMBERS!!!!')
     ILNUM=1000
     ILNUM=1000+4000
C
C     READ IN DISTANCES FROM M25B FROM UNIT 4
C
     READ (4,70) ALPHA
     READ (4,60) (A(I),I=1,160)
     READ (4,80) (DIST25(I),I=1,160)
  60 FORMAT (160A4)
  70 FORMAT (A4)
  80 FORMAT (160F10.4)
C
C   READ IN TRACE DATA
C
     DISTAV = 0.0
     DIST2 = 0.0
     WRITE (6,20)
  20 FORMAT (/'HOW MANY TRACES(I2)?'/'__')
     READ (5,30) N
  30 FORMAT (I2)
     WRITE (7,110)
```

```
 110 FORMAT (//'STATION DIST(M25B)  SHOT  DIST(M25B)  DIST. MIDPOINT(M2
    15B)   DIST'/)
     DO 130 I = 1, N
       READ (1'ILNUM ,40) IEVENT,ELAT,ELONG,EELEV,IEDAY,IEHOUR ,
    1  IEMIN   , ESEC   , ISTAT   , SLAT, SLONG, SELEV, KSDAY   ,
    2  KSHOUR   , KSMIN   , KSSEC   , JSAMP   , SAMRAT   , GAIN   ,
    3  IPOL   , DIST   , DAZ
  40   FORMAT (A4, 6X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, A4, 6X,
    1         3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, I6, F10.4, 10X,
    2         E10.4, 2X, A4, 2F10.3)
C
C     FIND DISTANCES OF EACH STATION AND SHOT FORM M25B
C
       WRITE(6,2) IEVENT,ISTAT
  2    FORMAT('PROCESSING  ',2A4)
       DO 90 II = 1, 160
         IF (A(II) .EQ. IEVENT   ) DISTSH    = DIST25(II)
         IF (A(II) .EQ. ISTAT   ) DISTST    = DIST25(II)
  90   CONTINUE
       DISTMP    = (DISTSH    + DISTST    ) / 2
       DISTAV = (DIST + DISTAV)
       DIST2 = DIST ** 2 + DIST2
       WRITE (7,120) ISTAT   , DISTST   , IEVENT   , DISTSH   ,
    1  DISTMP   , DIST
 120   FORMAT (2(A4,1X,F10.3,5X), F10.3, 7X, F10.3)
       WRITE (1'ILNUM,40) IEVENT, ELAT, ELONG, EELEV, IEDAY, IEHOUR ,
    1  IEMIN   , ESEC   , ISTAT   , SLAT, SLONG, SELEV, KSDAY   ,
    2  KSHOUR   , KSMIN   , KSSEC   , JSAMP   , SAMRAT   , GAIN   ,
    3  IPOL   , DISTMP   , DAZ
     ILNUM=ILNUM+2000
 130 CONTINUE
     DISTAV = DISTAV / N
     IF (N .EQ. 1) GO TO 140
     DISTSD = (DIST2 - N*(DISTAV**2)) / (N - 1)
     DISTSD = DISTSD ** 0.5
     GO TO 150
 140 DISTSD = 0.0
 150 WRITE (7,160) DISTAV, DISTSD
 160 FORMAT (/'AVERAGE DISTANCE IS ', F10.3, /'STANDARD DEV IS ',
    1       F10.4)
 170 STOP
     END
C
```

```
C
CA    ASP GREEN AND AHJ LEWIS
C
CD+
C            VERSION ON 25TH OCT 1985
C*********************************************
C
C                    PLOT
C                    ----
C                     BY
C
C     A. S. P. GREEN AND A.H.J.L. LEWIS
C
C
C    PROGRAM FOR PLOTTING REDUCED SECTIONS OF
C    SEISMIC REFRACTION DATA FROM DIGITAL TAPES WRITTEN
C    IN THE DURHAM SELF-DEFINING TAPE FORMAT (DSDTF)
C    AND COPIED FROM TAPE TO DISK USING PROGRAM
C    GPT9:TR-CSP.COM
C
C    $R PLOT.COM+*GHOST
C    UNIT 1   OUTPUT FROM GPT9:TR-CSP.COM
C    UNIT 8   OUTPUT OF HEADER BLOCK READ IN FROM UNIT 1
C    UNIT 7   OUTPUT CONTAINING SAMPLE NUMBERS AND TIMES
C             OF THE SECTION IF TRACE PLOTTED
C    UNIT 9   PLOTFILE
C
C    UNITS 5 & 6 ARE USED FOR INTERACTIVE INPUT OF DATA
C    IF THE SECTION IS REQUIRED UNREDUCED A REDUCTION
C    VELOCITY OF 0.0 SHOULD BE USED
C
C    THERE IS AN UPPER LIMIT OF 32768 BYTES(16384SAMPLES)
C    OF DATA TO EACH TRACE ON UNIT 1
C    THE MAXIMUM NO. OF TRACES(N) IS 90
C    ADDITIONAL RESTRICTIONS ARE:-
C      MAX. NO. OF SAMPLES PER TRACE = 126000/N
C      UPPER LIMIT OF 10000 SAMPLES
C
C    FILTERING OF TRACES GREATER THAN 4096 SAMPLES
C    IS NOT POSSIBLE
C
C    It is important to read the documentation attatched to the
C    deconvolution routines before attempting to deconvolve the
C    traces.
C
C    There is now an option to plot travel-times held in the travel-time
C    matrices Gpt9:tmat-Pg, etc....
C
C    IT HELPS THE AUTOMATIC AMPLITUDE ROUTINE IF
C    AT LEAST SOME OF THE TRACES ARE IN DISTANCE
C    ORDER. IT ONLY BREAKS DOWN IF TWO TRACES HAVE
C    THE SAME OR VERY NEARLY THE SAME RANGE
C
C    X(A,B)    - ARRAY WITH THE AMPLITUDE OF SAMPLES
C    TSAM(A,B) - ARRAY WITH THE TIME OF SAMPLES
```

```
C            A - TRACE NUMBER
C            B - SAMPLE NUMBER
C
C    N          - NUMBER OF TRACES
C    VRED       - REDUCTION VELOCITY (KM/S)
C    TLEN       - LENGTH OF PLOT (SECONDS)
C    TSRED      - REDUCED START TIME OF PLOT (SECONDS)
C
C    THE PLOT DISTANCES ARE CALCULATED BY GPT9:TR-CSP.COM
C    FROM THE SHOT AND STATION LAT/LONG ON THE DSDTF TAPE
C    USING A SUBROUTINE BASED ON THE PROGRAM DISTAZ BY
C    G. K. WESTBROOK.
C    THIS PROGRAM USES THE SIGN OF THE AZIMUTH TO PLOT
C    WEST TO THE RIGHT AND EAST TO THE LEFT. IF ALL PLOTS
C    ARE REQUIRED WITH INCREASING RANGE TO THE RIGHT THE
C    AZIMUTH MUST BE POSITIVE.
C
C
C*************************************************
C
C
C    THE MAIN ROUTINE EXISTS ONLY TO MAKE THE TRACE
C    ARRAY, X, INTO A 2-D ARRAY WHOSE DIMENSIONS ARE
C    DETERMINED BY THE NUMBER OF TRACES
C
C
      REAL*4 X(126000), TSAM(126000)
      WRITE (6,10)
   10 FORMAT ('NO. OF TRACES'/'.....    I5')
      READ (5,20) N
   20 FORMAT (I5)
   30 NARG = 126000 / N
      CALL MAINP(N, NARG, X, TSAM)
      STOP
      END
C
C
C
      SUBROUTINE MAINP(N, NARG, X, TSAM)
C
      DIMENSION PROJ(20), TITLE(20), STAR(20), IEVENT(90,5),
     1          ISTAT(90,5), DUMP(20), IEVEN(5), ISTA(5), XP(2000),
     2          TP(2000)
      REAL*4 ESEC(90), DIST(90), SAMRAT(90), XTEMP(10000), KSSEC(90),
     1       GAIN(90), TSAM1(10000), TSAM(N,NARG), X(N,NARG)
      REAL*8 DTEVE(90), DTST(90)
      INTEGER*4 IEHOUR(90), IEMIN(90), NST(90), NEND(90), KSHOUR(90),
     1          KSMIN(90), LX(90), IPOL(90), JSAMP(90), KSDAY(90),
     2          IEDAY(90), ITIME(5), BLANK
      INTEGER*2 LEN, ISAM(16500)
      DATA BLANK /'    '/
      DATA M /'M   '/
C
C   READ TITLE
C
      WRITE (6,10)
   10 FORMAT ('PROJECT TITLE')
      READ (5,60) PROJ
      WRITE (6,20)
   20 FORMAT ('SECTION TITLE')
      READ (5,60) TITLE
C
C   READ IN PLOTTING PARAMETERS
C
      WRITE (6,30)
   30 FORMAT ('TRACE LENGTH(SECS) RED.VEL(KM/SEC) AND START TIME'/
     1        '(UNREDUCED SECTION RED.VEL=0.0)'/'.....———.....  3F5.2')
      READ (5,40) TLEN, VELR, TSRED
   40 FORMAT (3F5.2)
      WRITE (6,50)
   50 FORMAT ('SECTION TYPE, COM. EVENT=1, COM. STAT.=2, OTHER=0')
      READ (5,240) ITYPE
C
C   READ HEADING
C
      READ (1,60) STAR
      READ (1,60) DUMP
      READ (1,60) STAR
      READ (1,60) DUMP
   60 FORMAT (20A4)
C
      WRITE (7,70)
   70 FORMAT ('FILE START    EVENT START   1ST SAMPLE  LAST SAMPLE   PLO
     1T START   RED. PLOT START   RED. PLOT END')
      WRITE (7,80)
   80 FORMAT ('TIME (SECS)   TIME (SECS)    PLOTTED      PLOTTED     TIM
     1E (SECS)    TIME (SECS)      TIME (SECS)'/)
C
C   READ IN TRACE DATA
C
      DO 120 I = 1, N
        READ (1,90) (IEVENT(I,J),J=1,2), ELAT, ELONG, EELEV, IEDAY(I),
     1  IEHOUR(I), IEMIN(I), ESEC(I), (ISTAT(I,J),J=1,2), SLAT, SLONG,
     2  SELEV, KSDAY(I), KSHOUR(I), KSMIN(I), KSSEC(I), JSAMP(I),
     3  SAMRAT(I), GAIN(I), IPOL(I), DIST(I), DAZ
        WRITE (8,90) (IEVENT(I,J),J=1,2), ELAT, ELONG, EELEV, IEDAY(I),
     1  IEHOUR(I), IEMIN(I), ESEC(I), (ISTAT(I,J),J=1,2), SLAT, SLONG,
     2  SELEV, KSDAY(I), KSHOUR(I), KSMIN(I), KSSEC(I), JSAMP(I),
     3  SAMRAT(I), GAIN(I), IPOL(I), DIST(I), DAZ
   90   FORMAT (2A4, 2X, 3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, 2A4, 2X,
     1         3F12.6, I4, 1X, I2, 2X, I2, F8.4, 2X, I6, F10.4, 10X,
     2         E10.4, 2X, A4, 2F10.3)
        IF (ITYPE .EQ. 1 .AND. DAZ .LT. 0.0) DIST(I) = -DIST(I)
        IF (ITYPE .EQ. 2 .AND. DAZ .GT. 0.0) DIST(I) = -DIST(I)
        CALL READ(ISAM, LEN, 0, LNUM, 1, &640)
        LENG = LEN / 2
C
C   CONVERT FILE START TIME AND SHOT TIME TO SECONDS
```

```
C
      CALL SECOND(KSHOUR, KSMIN, KSSEC, DTST, I)
      CALL SECOND(IEHOUR, IEMIN, ESEC, DTEVE, I)
C
C   DETERMINE WHICH PART OF THE DATA TO PLOT
C
      CALL SAMPLE(DTST, DTEVE, TSRED, DIST, NST, SAMRAT, NEND, TLEN,
     1        I, VELR, TSAM, NARG, N)
      LX(I) = NEND(I) - NST(I) + 1
      IF (LX(I) .LE. NARG) GO TO 100
      NEND(I) = NST(I) + NARG - 1
      LX(I) = NARG
 100  K = 1
      NS = NST(I)
      NE = NEND(I)
      DO 110 J = NS, NE
        X(I,K) = ISAM(J)
        K = K + 1
 110  CONTINUE
 120 CONTINUE
C
C   MEAN CORRECT
C
     CALL MEAN(X, N, LX, NARG)
C
C   PREDICTIVE DECON CALCULATION
C
     WRITE (6,130)
 130 FORMAT ('DECONVOLUTION  YES = 1  NO = 0 & NUM OF ITERATIONS'/
     1        '_——')
     READ (5,140) NI, NUMDEC
     IF (NI .EQ. 0) GO TO 150
     CALL DECON(X, N, LX, NARG, ISTAT, NUMDEC)
 140 FORMAT (I1, I2)
C
C   BANDPASS FILTER
C
 150 WRITE (6,160)
 160 FORMAT ('FILTERING  YES = 1  NO = 0')
     READ (5,240) NI
     IF (NI .NE. 1) GO TO 221
     WRITE (6,170)
 170 FORMAT ('ENTER F1,F2,F3,F4'/'.....———.....———   5F5.2')
     READ (5,180) F1, F2, F3, F4
 180 FORMAT (5F5.2)
     DO 220 J = 1, N
       LXN = LX(J)
       WRITE (7,190) LXN
 190   FORMAT (I5)
       SAMRT = SAMRAT(J)
       DO 200 I = 1, LXN
         XTEMP(I) = X(J,I)
 200   CONTINUE
       CALL BPASS(LXN, F1, F2, F3, F4, XTEMP, SAMRT)
       DO 210 I = 1, LXN
         X(J,I) = XTEMP(I)
 210   CONTINUE
 220 CONTINUE
221    CONTINUE
C
C   ADJUST AMPLITUDE
C
     WRITE (6,230)
 230 FORMAT ('UNC. AMP. PLOT = 0 CORR.AMP. PLOT = 1 OR EQUAL.PLOT = 2')
     READ (5,240) NOR
 240 FORMAT (I1)
     IF (NOR .EQ. 0) GO TO 261
     IF (NOR .EQ. 1) GO TO 251
     CALL NORM(X, N, LX, NARG)
     GO TO 261
 250 CONTINUE
251  DO 260 I = 1, N
       IF (GAIN(I) .NE. 0.001221) CALL AMP(X, I, N, LX, GAIN, NARG)
 260 CONTINUE
261         CONTINUE
C
C   CHECK AND CHANGE POLARITY
C
     DO 270 I = 1, N
       CALL POLAR(X, I, N, LX, NARG, ISTAT)
 270 CONTINUE
C
C   READ IN PLOT SCALE
C
     KP = 0
 280 WRITE (6,290)
 290 FORMAT ('TIMESCALE(CM/SEC),DISTANCE(CM/KM),RETURN FOR DEFAULT
    1  '/'.....———     2F5.2')
     READ (5,300) TSCALE, DSCALE
 300 FORMAT (2F5.2)
     IF (TSCALE .EQ. 0.0) TSCALE = 2.0
     IF (DSCALE .EQ. 0.0) DSCALE = 0.4
C
     IF (KP .EQ. 1) GO TO 310
C
C
     CALL AMPFAK(X, N, LX, DIST, NOR, NARG, TSCALE, AMARG, AMPFAC)
     AMARG = AMARG * 1.5
C
C   PLOT AXES
C
 310 DMIN = 999.0
     DMAX = -999.0
     DO 320 I = 1, N
       IF (DMAX .LT. DIST(I)) DMAX = DIST(I)
       IF (DMIN .GT. DIST(I)) DMIN = DIST(I)
 320 CONTINUE
     TMIN = TSRED
     TMAX = TSRED + TLEN
     CALL PAPER(1)
```

```
      T = (TMAX - TMIN) * TSCALE
  330 D = (DMAX - DMIN + (2*AMARG)) * DSCALE
      CALL PSPACE(0.1, D/25.4 + 0.1, 0.1, T/25.4 + 0.1)
      CALL CSPACE(0.0, D/25.4 + 0.1 + 0.45, 0.0, T/25.4 + TMAX/10.0)
      CALL MAP(DMIN - AMARG, DMAX + AMARG, TMIN, TMAX)
      CALL CTRMAG(8)
      SY = 1.0
      SX = 5.0
      IF (DSCALE .GT. 2.0) SX = 0.5
      IF (TSCALE .GT. 8.0) SY = 0.5
      CALL SCALSI(SX, SY)
C
C   LABEL AXES
C
      DX = DMIN + (DMAX - DMIN) / 2.3
      TX = TMIN - 1 / TSCALE
      CALL PLOTCS(DX, TX, 'DISTANCE(KM)', 13)
      DX = DMIN + (DMAX - DMIN) * 4.0 / 5.0
      CALL TIME(6, 0, ITIME)
      CALL PLOTCS(DX, TX, ITIME, 20)
      DX = DMIN + (DMAX - DMIN) / 5.0
      IF (ITYPE .EQ. 0) GO TO 350
      IF (ITYPE .EQ. 1) GO TO 340
      CALL PLOTCS(DMIN, TX, 'CSSP COMMON STATION ', 20)
      CALL PLOTCS(DX, TX, ISTAT, 4)
      GO TO 360
  340 CALL PLOTCS(DMIN, TX, 'CSSP COMMON SHOT ', 17)
      CALL PLOTCS(DX, TX, IEVENT, 4)
      GO TO 360
  350 CALL PLOTCS(DMIN, TX, TITLE, 30)
      DX = DMIN + (DMAX - DMIN) / 6.0
      CALL PLOTCS(DX, TX, ISTAT, 4)
      DX = DMIN + (DMAX - DMIN) / 4.5
      CALL PLOTCS(DX, TX, IEVENT, 4)
      GO TO 360
  360 CALL CTRORI(3.0)
      TX = TMAX
      IF (NI .EQ. 0) GO TO 370
      CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE, TX, 'FILTERED', 8)
      CALL PLOTNF(DMAX + AMARG + 1.1/DSCALE, TX - 1.9/TSCALE, F2, 1)
      CALL PLOTNF(DMAX + AMARG + 1.1/DSCALE, TX - 3.0/TSCALE, F3, 1)
  370 IF (NI .EQ. 0) CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE,
     1     TX - 1.4/TSCALE, 'UNFILTERED', 10)
      IF (NOR .EQ. 2) GO TO 380
      IF (NOR .EQ. 0) GO TO 390
      CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE, TX - 9.5/TSCALE,
     1     'TRACE AMPFAC', 12)
      CALL PLOTNF(DMAX + AMARG + 1.1/DSCALE, TX - 12.5/TSCALE, AMPFAC,
     1     7)
      GO TO 400
  380 CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE, TX - 11.5/TSCALE,
     1     'EQUALIZED', 9)
      GO TO 400
  390 CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE, TX - 9.5/TSCALE,
     1     'UNCORR AMPFAC ', 14)
      CALL PLOTNF(DMAX + AMARG + 1.1/DSCALE, TX - 12.5/TSCALE, AMPFAC,
     1     7)
  400 CALL PLOTCS(DMAX + AMARG + 1.1/DSCALE, TX - 4.0/TSCALE,
     1     'TIME(SECS) RED TO      KM/SEC', 29)
      CALL PLOTNF(DMAX + AMARG + 1.1/DSCALE, TX - 7.4/TSCALE, VELR, 1)
      CALL CTRORI(0.0)
C
C   PLOT TRACES
C
      DO 450 I = 1, N
        CALL MAP(DMIN - AMARG - DIST(I), DMAX + AMARG - DIST(I), TMIN,
     1       TMAX)
        NUM = LX(I)
        DO 410 K = 1, NUM
          XTEMP(K) = X(I,K)
          TSAM1(K) = TSAM(I,K)
  410   CONTINUE
        CALL PTPLOT(XTEMP, TSAM1, 1, NUM, -2)
        DO 420 K = 1, 5
          IEVEN(K) = IEVENT(I,K)
          ISTA(K) = ISTAT(I,K)
  420   CONTINUE
        IF (ITYPE .EQ. 2) GO TO 430
        IF (ITYPE .EQ. 1) GO TO 440
        CALL PLOTCS(0, TMAX + 1.3/TSCALE, ISTA, 4)
  430   CALL PLOTCS(0, TMAX + 0.5/TSCALE, IEVEN, 4)
        GO TO 450
  440   CALL PLOTCS(0, TMAX + 0.5/TSCALE, ISTA, 4)
  450 CONTINUE
C
C     INPUT TRAVEL TIMES FROM MATRIX FILES
C
      WRITE (6,460)
  460 FORMAT (/'DO YOU WANT TO PLOT TRAVEL TIMES FROM UNIT 2 ? N=0'/,
     1        ' IF YES IO=  1=CST,2=CSH,3=CDS,4=OMP PLOT', /
     2        'INPUT IO,SEA(M OR N), STAT, SHOT, CDS DISTANCE'/
     3        '____________________________________')
      READ (5,470) IO, ISEA, IST, ISH, XCDS
  470 FORMAT (I2, 2X, 3A4, F10.3)
      IF (IO .EQ. 0) GO TO 510
      CALL MAP(DMIN - AMARG, DMAX + AMARG, TMIN, TMAX)
      CALL TTREAD(XP, TP, NUMP, IO, ISEA, IST, ISH, XCDS)
      WRITE (7,480) NUMP
  480 FORMAT (/'TT READ FROM MATRIX FILES - NUM=', I5)
      DO 500 I = 1, NUMP
        IF (ITYPE .EQ. 1 .AND. DAZ .LT. 0.0) XP(I) = -XP(I)
        IF (ISEA .EQ. M .AND. IO.EQ.1) XP(I) = -XP(I)
        IF (ITYPE .EQ. 2 .AND. DAZ .GT. 0.0) XP(I) = -XP(I)
        IF (ITYPE .GE. 3 .AND. DIST(N).LT.0.0) XP(I) = -XP(I)
        IF (VELR .NE. 0.0) TP(I) = TP(I) - ABS(XP(I)) / VELR
        WRITE (7,490) XP(I), TP(I)
  490   FORMAT (12X, 2F10.3)
  500 CONTINUE
      CALL CTRMAG(25)
      CALL PTPLOT(XP, TP, 1, NUMP, 44)
```

```
 510 CALL BORDER
C
C   WRITE PLOT PARAMETERS
C
     WRITE (6,520)
 520 FORMAT (/'DO YOU WANT A HEADER BLOCK FOR PLOT ? YES=0 NO=1')
     READ (5,530) IQ
 530 FORMAT (I1)
     IF (IQ .EQ. 1) GO TO 600
     CALL PSPACE(D/25.4 + 0.1 + 0.02, D/25.4 + 0.1 + 0.36, 0.1, T/25.4
    1     + 0.1)
     CALL MAP(0.0, 10.0, 0.0, 10.0)
     CALL BORDER
     CALL CTRMAG(15)
     CALL PLOTCS(1.0, 9.0, PROJ, 30)
     CALL PLOTCS(1.0, 8.0, TITLE, 30)
     CALL CTRMAG(12)
     IF (ITYPE .EQ. 0) GO TO 550
     IF (ITYPE .EQ. 1) GO TO 540
     CALL PLOTCS(1.0, 6.5, 'COMMON STATION SECTION', 22)
     CALL PLOTCS(1.5, 6.0, 'STATION', 7)
     CALL PLOTCS(4.0, 6.0, ISTAT, 4)
     GO TO 550
 540 CALL PLOTCS(1.0, 6.5, 'COMMON EVENT SECTION', 20)
     CALL PLOTCS(1.5, 6.0, 'SHOT', 4)
     CALL PLOTCS(4.0, 6.0, IEVENT, 4)
 550 CALL CTRMAG(10)
     IF (NI .EQ. 0) GO TO 560
     CALL PLOTCS(1.0, 4.5, 'FILTER(HZ)', 10)
     CALL PLOTNF(4.0, 4.5, F2, 1)
     CALL PLOTNF(5.5, 4.5, F3, 1)
     GO TO 570
 560 CALL PLOTCS(1.0, 4.5, 'FILTER  -  NONE', 15)
 570 CALL PLOTCS(1.0, 3.5, 'AMPLITUDE', 9)
     IF (NOR .EQ. 2) GO TO 580
     IF (NOR .EQ. 0) GO TO 590
     CALL PLOTCS(1.5, 3.0, 'CORRECTED TO COMMON GAIN', 24)
     GO TO 600
 580 CALL PLOTCS(1.5, 3.0, 'EQUALISED', 9)
     GO TO 600
 590 CALL PLOTCS(1.5, 3.0, 'UNADJUSTED', 10)
 600 CONTINUE
C
C   ASK IF PLOT WANTED AT ANOTHER SCALE
C
     WRITE (6,610)
 610 FORMAT ('DO YOU WANT THE SAME PLOT AT ANOTHER SCALE Y=1 N=0')
     READ (5,620) KP
 620 FORMAT (I1)
     IF (KP .EQ. 0) GO TO 630
     CALL FRAME
     GO TO 280
 630 CALL GREND
 640 RETURN
     END
C
C*********************************************
C
C        SUBROUTINE SECOND
C        ------------------
C
C    CONVERTS THE TIME IN HOURS MINS AND  SECONDS
C    TO SECONDS
C
C*********************************************
C
     SUBROUTINE SECOND(IHOUR, IMIN, DSECS, DTIME, I)
     REAL*4 DSECS(90)
     REAL*8 DTIME(90), DATIME
     INTEGER*4 IHOUR(90), IMIN(90)
C
     ITIME = (IHOUR(I)*3600) + (IMIN(I)*60)
     DATIME = ITIME
     DTIME(I) = DATIME + DSECS(I)
     RETURN
     END
C
C*********************************************
C
C        SUBROUTINE MEAN
C        ---------------
C
C    MEAN CORRECTS THE DATA
C
C*********************************************
C
     SUBROUTINE MEAN(X, N, LX, NARG)
     REAL*4 X(N,NARG)
     INTEGER*4 LX(N)
C
     DO 30 J = 1, N
       XTOT = X(J,1)
       LEN = LX(J)
       DO 10 I = 2, LEN
         XTOT = XTOT + X(J,I)
  10   CONTINUE
       XAV = XTOT / LEN
       DO 20 I = 1, LEN
         X(J,I) = X(J,I) - XAV
  20   CONTINUE
  30 CONTINUE
     RETURN
     END
C
C*********************************************
C
C        SUBROUTINE NORM
C        ---------------
C
C   NORMALISES THE SEISMIC TRACES TO 1.0
```

```
C
C*************************************************
C
      SUBROUTINE NORM(X, N, LX, NARG)
      REAL*4 X(N,NARG)
      INTEGER*4 LX(N)
C
      DO 50 I = 1, N
        XMAX = -1000.0
        LEN = LX(I)
        DO 10 J = 1, LEN
          Y = ABS(X(I,J))
          IF (Y .GT. XMAX) XMAX = Y
   10   CONTINUE
        IF (XMAX .EQ. 0.0) GO TO 30
        DO 20 J = 1, LEN
          X(I,J) = X(I,J) / XMAX
   20   CONTINUE
        GO TO 50
   30   WRITE (6,40) I
   40   FORMAT (1X, 'MAXAMP=0-NORMALIZATION NOT POSSIBLE,TRACE NO=', I6)
   50 CONTINUE
      RETURN
      END
C
C*************************************************
C
C        SUBROUTINE SAMPLE
C        ------------------
C
C     CALCULATES THE FIRST AND LAST SAMPLES REQUIRED
C     FROM THE TRACE ARRAYS FOR PLOTTING
C       VELR     -     RED. VEL.
C       DTEVE    -     SHOT TIME
C       DTST     -     FILE START TIME
C       TSRED    -     START TIME(RED) OF PLOT WANTED
C       TSAM     -     TIME(RED) OF SAMPLES TO BE PLOTTED
C       TLEN     -     LENGTH(RED.TIME) OF PLOT
C       NST      -     1st SAMPLE TO BE PLOTTED
C       NEND     -     LAST SAMPLE TO BE PLOTTED
C
C*************************************************
C
      SUBROUTINE SAMPLE(DTST, DTEVE, TSRED, DIST, NST, SAMPR, NEND,
     1            TLEN, I, VELR, TSAM, NARG, N)
      REAL*4 DIST(90), SAMPR(90), TSAM(N,NARG), VELR
      REAL*8 DTPLOT(90), DSAMNO, DSAM, DTST(90), DTEVE(90)
      INTEGER*4 NST(90), NEND(90)
C
      TCORR = 0.0
      IF (VELR  EQ. 0.0) GO TO 10
      DTPLOT(I) = ABS(DIST(I)) / VELR + DTEVE(I) + TSRED
      GO TO 20
   10 DTPLOT(I) = DTEVE(I) + TSRED
   20 IF (DTPLOT(I)  GE. DTST(I)) GO TO 30
      TCORR = DTST(I) - DTPLOT(I)
      DTPLOT(I) = DTST(I)
   30 DSAMNO = (DTPLOT(I) - DTST(I)) * SAMPR(I) + 1.0
      NST(I) = IDINT(DSAMNO) + 1
      TSAM(I,1) = TSRED + TCORR + (NST(I) - DSAMNO) / SAMPR(I)
      DSAM = 1.0 / SAMPR(I)
      DO 50 J = 2, 10000
        K = J - 1
        TSAM(I,J) = TSAM(I,K) + DSAM
   40   IF (TSAM(I,J) .GT. (TSRED + TLEN)) GO TO 60
   50 CONTINUE
   60 NEND(I) = NST(I) + K
      WRITE (7,70) DTST(I), DTEVE(I), NST(I), NEND(I), DTPLOT(I),
     1TSAM(I,1), TSAM(I,K)
   70 FORMAT (F10.3, 4X, F10.3, 5X, I5, 8X, I5, 7X, F10.3, 4X, F9.3, 8X,
     1        F9.3)
      NDIFF = NEND(I) - NST(I)
      IF (NDIFF .GT. NARG) NEND(I) = NARG + NST(I)
   80 RETURN
      END
C
C*************************************************
C
C        SUBROUTINE POLAR
C        ----------------
C
C     MULTIPLIES THE SIGNAL BY -1.0
C
C*************************************************
C
      SUBROUTINE POLAR(X, I, N, LX, NARG, ISTAT)
      REAL*4 X(N,NARG)
      INTEGER*4 LX(N), ISTAT(N,5), NSTAT(5)
      DATA NSTAT /'23  ', '31  ', '36  ', '53  ', '56  '/
C
      NST = 0
      DO 10 J = 1, 5
   10 IF (ISTAT(I,1) .EQ. NSTAT(J)) NST = 1
      IF (NST .EQ. 0) RETURN
      LXN = LX(I)
      DO 20 J = 1, LXN
        X(I,J) = -X(I,J)
   20 CONTINUE
      RETURN
      END
C
C*************************************************
C
C        SUBROUTINE AMP
C        --------------
C
C     CONVERTS SIGNALS TO SAME GAIN
C     BASED ON GAIN 7 FOR THE AMPLIFIER MODULATOR UNITS
C     FROM THE NERC SEISMIC EQUIPMENT POOL
C     TO CHANGE TO ANOTHER BASE CHANGE THE VALUE
```

```
C     0.001221. IT ONLY OCCURRS TWICE, IN THIS SUBROUTINE
C     AND ONCE IN THE SUBROUTINE CALL
C
C*************************************************
C
      SUBROUTINE AMP(X, J, N, LX, GAIN, NARG)
      REAL*4 X(N,NARG), GAIN(N)
      INTEGER*4 LX(N)
C
      FAC = GAIN(J) / 0.001221
      WRITE (6,10) FAC, GAIN(J)
   10 FORMAT (/'FAC,TRACE PLOTTING GAIN= ', 2E10.4)
      LENG = LX(J)
      DO 20 I = 1, LENG
        X(J,I) = X(J,I) * FAC
   20 CONTINUE
      RETURN
      END
C
C*************************************************
C
C        SUBROUTINE AMPFAK
C
C
C     ADJUSTS THE AMPLITUDES FOR PLOTTING
C
C*************************************************
C
      SUBROUTINE AMPFAK(X, N, LX, DIST, NCR, NARG, TSCALE, DMIN, AMPFAC)
C
      REAL*4 X(N,NARG), DIST(N), XMAX(90), XMIN(90)
      INTEGER*4 LX(N)
C
      IF (NCR .EQ. 2) GO TO 40
C
      DO 20 I = 1, N
        NUM = LX(I)
        XMAX(I) = -1000.0
        XMIN(I) = 1000.0
        DO 10 K = 1, NUM
          IF (XMAX(I) .LT. X(I,K)) XMAX(I) = X(I,K)
          IF (XMIN(I) .GT. X(I,K)) XMIN(I) = X(I,K)
   10   CONTINUE
   20 CONTINUE
C
      MAX = 0.0
      MIN = 0.0
      DO 30 I = 1, N
        MAX = MAX + XMAX(I)
        MIN = MIN + XMIN(I)
   30 CONTINUE
C
      AVMAX = MAX / N
      AVMIN = MIN / N
      AVER = ABS(AVMAX) + ABS(AVMIN)
      GO TO 50
   40 AVER = 1.5
      IF (TSCALE .GT. 8.0) AVER = 0.5
C
   50 WRITE (6,60) AVER
   60 FORMAT ('AMPFAC = ', F15.3)
      WRITE (6,70)
   70 FORMAT ('ENTER ALTERNATIVE AMPFAC (FOR DEFAULT VALUE ABOVE,RETURN)
     1'/'A SMALLER VALUE GIVES A PROPORTIONALLY BIGGER AMPLITUDE'/
     2        '........    F8.3')
      READ (5,80) AV
   80 FORMAT (F8.3)
      IF (AV .NE. 0.0) AVER = AV
      DMIN = 100.0
      IF (N .EQ. 1) GO TO 100
      DO 90 I = 2, N
        IF (ABS(DIST(I) - DIST(I - 1)) .LT. DMIN) DMIN = ABS(DIST(I) -
     1  DIST(I - 1))
   90 CONTINUE
      GO TO 110
  100 DMIN = 5.0
C
  110 DO 130 I = 1, N
        NUM = LX(I)
        DO 120 K = 1, NUM
          X(I,K) = (X(I,K)*DMIN) / AVER
  120   CONTINUE
  130 CONTINUE
      AMPFAC = DMIN / AVER
      RETURN
      END
C
C*****************************************************************
C      SUBROUTINE BPASS1
C
C
C    AN EDITED VERSION OF BRIAN RUSSELL'S SUBROUTINE TO
C    PERFORM BANDPASS FILTERING ON EACH SEISMIC TRACE.
C    SUBROUTINES FFTA SINTAB & HANN ARE CALLED.
C
C    X = INPUT ARRAY
C    LX = LENGTH OF X
C    SAMPR = SAMPLING RATE
C    F1,F2,F3,F4 = DEFINES FREQUENCY RANGE FOR HANNING WINDOW
C
C    F1 MUST BE GREATER THAN THE FUNDAMENTAL FREQUENCY
C    F4 MUST BE LESS THAN THE NYQUIST FREQUENCY
C
C*****************************************************************
C
      SUBROUTINE BPASS(LX, F1, F2, F3, F4, X, SAMPR)
      DIMENSION X(4096), Y(4096), S(4096), F(4096)
C
      LEN = LX
      CALL PAD(LEN, X)
```

```
      L1 = LEN / 2 + 1
      L2 = L1 + 1
      DO 10 I = 1, LEN
        S(I) = 0.0
        Y(I) = 0.0
   10 CONTINUE
C
C   TRANSFORM THE TIME SERIES
C
      CALL SINTAB(LEN, S)
      SIGNI = -1.0
      CALL FFTA(LEN, X, Y, SIGNI, S)
      DF = SAMPR / DFLOAT(LEN)
      DO 20 I = 1, L1
        F(I) = DF * DFLOAT(I)
   20 CONTINUE
C
C   SET UP SAMPLE NUMBERS FOR HANNING WINDOW LIMITS
C
      DO 30 I = 2, L1
        IF (F1 .GE. F(I - 1) .AND. F1 .LE. F(I)) JJ = I - 1
        IF (F2 .GE. F(I - 1) .AND. F2 .LE. F(I)) J = I - 1
        IF (F3 .GE. F(I - 1) .AND. F3 .LE. F(I)) K = I - 1
        IF (F4 .GE. F(I - 1) .AND. F4 .LE. F(I)) KK = I - 1
   30 CONTINUE
      CALL HANN(X, L1, JJ, KK, J, K)
      CALL HANN(Y, L1, JJ, KK, J, K)
      DO 40 I = L2, LEN
        X(I) = X(LEN - I + 2)
        Y(I) = -Y(LEN - I + 2)
   40 CONTINUE
      SIGNI = 1.0
      CALL FFTA(LEN, X, Y, SIGNI, S)
      RETURN
      END
C
C*****************************************************************
C
C     SUBROUTINE FFTA
C     ---------------
C
C   DESCRIPTION - THIS SUBROUTINE COMPUTES THE FAST FOURIER
C                 TRANSFORM BY THE COOLEY-TUKEY ALGORITHM.
C   PARAMETERS - LX=NUMBER OF DATA POINTS
C                X=ARRAY CONTAINING REAL PART OF TRANSFORM
C                Y=ARRAY CONTAINING IMAGINARY PART OF TRANSFORM
C                SIGNI=+1. FOR FORWARD TRANSFORM
C                     =-1. FOR REVERSE TRANSFORM
C                S=ARRAY OF SINE VALUES GENERATED BY SUBROUTINE
C                  SINTAB CALLED BY FFTA
C
C   SOURCE - CLAERBOUT, MODIFIED BY ALAN NUNNS
C*****************************************************************
C
      SUBROUTINE FFTA(LX, X, Y, SIGNI, S)
      DIMENSION X(LX), Y(LX), S(LX)
      SC = 1.0
      IF (SIGNI .LT. 0.0) SC = 1.0 / LX
      NN = LX / 2
      N = LX / 4
      J = 1
      DO 30 I = 1, LX
        IF (I .GT. J) GO TO 10
        TEMPX = X(J) * SC
        TEMPY = Y(J) * SC
        X(J) = X(I) * SC
        Y(J) = Y(I) * SC
        X(I) = TEMPX
        Y(I) = TEMPY
   10   M = NN
   20   IF (J .LE. M) GO TO 30
        J = J - M
        M = M / 2
        IF (M .GE. 1) GO TO 20
   30 J = J + M
      L = 1
      NL = NN
   40 ISTEP = 2 * L
      IND = 1
      DO 100 M = 1, L
        INDN = IND - N - 1
        IF (INDN) 50, 60, 70
   50   WX = S(1 - INDN)
        WY = S(IND) * SIGNI
        GO TO 80
   60   WX = 0.
        WY = SIGNI
        GO TO 80
   70   WX = -S(INDN + 1)
        WY = S(N + 1 - INDN) * SIGNI
   80   DO 90 I = M, LX, ISTEP
          TEMPX = WX * X(I + L) - WY * Y(I + L)
          TEMPY = WX * Y(I + L) + WY * X(I + L)
          X(I + L) = X(I) - TEMPX
          Y(I + L) = Y(I) - TEMPY
          X(I) = X(I) + TEMPX
   90   Y(I) = Y(I) + TEMPY
  100 IND = IND + NL
      L = ISTEP
      NL = NL / 2
      IF (L .LT. LX) GO TO 40
      RETURN
      END
C
C*****************************************************************
C
C     SUBROUTINE SINTAB
C     -----------------
C
C   DESCRIPTION - THIS SUBROUTINE RETURNS VALUES
```

```
C   S(I)=SIN(2*PI*(I-1/)/LX)  I=1,2,....,LX/4+1
C   WHERE LX=2**INTEGER IS THE NUMBER OF POINTS IN THE
C   TIME SERIES IN MAIN
C
C   WRITTEN BY ALAN NUNNS
C*************************************************************
C
      SUBROUTINE SINTAB(LX, S)
      DIMENSION S(LX)
      DOUBLE PRECISION ARG, DELARG
      N1 = LX / 4 + 1
      ARG = 0.0
      DELARG = 0.6283185307179586D01 / LX
      DO 10 I = 1, N1
        S(I) = DSIN(ARG)
        ARG = ARG + DELARG
   10 CONTINUE
      RETURN
      END
C
C*************************************************************
C
C     SUBROUTINE HANN
C   ___________________
C   AN EDITED VERSION OF A SUBROUTINE BY BRIAN RUSSELL
C   IT APPLIES A HANNING WINDOW TO THE TRANSFORMED DATA
C   BETWEEN THE FREQUENCIES SET UP IN BPASS
C
C   X  = SPECTRUM OF DATA
C   N  = SIZE OF ARRAY
C   JJ = ELEMENT OF ARRAY STARTING WINDOW
C   KK = ELEMENT OF ARRAY ENDING THE WINDOW
C   J  = ELEMENT OF ARRAY ENDING RISING TAPER
C   K  = ELEMENT OF ARRAY STARTING DECREASING TAPER
C
C   VALUE OF WINDOW BETWEEN J & K = 1
C   WINDOW = 0 AT JJ & KK
C
C*************************************************************
C
      SUBROUTINE HANN(X, N, JJ, KK, J, K)
      DIMENSION X(N)
      NN = N - 1
      PI2 = 0.6283185307179 6D+01
      IF (JJ .EQ. 1) GO TO 20
      DO 10 I = 2, JJ
   10 X(I - 1) = X(I - 1) * 0.0D0
   20 IF (J .EQ. JJ) GO TO 40
      CON1 = PI2 / DFLOAT(2*(J - JJ))
      DO 30 I = JJ, J
        ARG = DFLOAT(I - JJ) * CON1
   30 X(I) = X(I) * 0.5D0 * (.1D+01 - COS(ARG))
   40 IF (K  EQ. KK) GO TO 60
      CON2 = PI2 / DFLOAT(2*(KK - K))
      DO 50 I = K, KK
        ARG = DFLOAT(KK - I) * CON2
   50 X(I) = X(I) * 0.5D0 * (.1D+01 - COS(ARG))
   60 IF (KK .EQ. N) GO TO 80
      DO 70 I = KK, NN
   70 X(I + 1) = X(I + 1) * 0.0D0
   80 RETURN
      END
C
C*************************************************************
C
C     SUBROUTINE PAD
C   ___________________
C
C   CHECKS TO SEE IF THE DATA LENGTH IS 2**N AND IF
C   NOT PADS THE ARRAY WITH ZEROS
C
C   X   =  DATA ARRAY
C   LX  =  LENGTH OF ARRAY
C
C*************************************************************
C
      SUBROUTINE PAD(LX, X)
      REAL*4 X(4096)
      DO 10 I = 1, 12
        IF (LX .EQ. 2**I) GO TO 50
        IF (LX .LT. 2**I) GO TO 20
   10 CONTINUE
   20 IDAT = 2 ** I
      WRITE (6,30) IDAT
   30 FORMAT ('TRACE PADDED WITH ZEROS TO ', I4, ' POINTS')
      LXN = IDAT
      LXT = LX + 1
      DO 40 I = LXT, LXN
        X(I) = 0.0
   40 CONTINUE
      LX = LXN
   50 RETURN
      END
C
C
C*********************                    ***********************
C                       PREDICTIVE DECONVOLUTION
C*********************                    ***********************
C
C Subroutine name : DECON
C
C Description :
C             This subroutine calls the various routines
C             required to deconvolve the traces using the
C             parametres specified in the input file.

      SUBROUTINE DECON(X, N, LX, NARG, ISTAT, NUMDEC)
C     ___________________________________________________
C
C
```

```
      IMPLICIT REAL*8(A - H,O - Z)
      REAL*4 X(N,NARG)
      INTEGER TNUM, ISTAT(90,5)
      DIMENSION TRACE(2048), SPACE1(2048), SPACE2(2048), SPACE3(2048),
     1          FILTER(2048), BUFFER(1024), LX(90)
C
      WRITE (6,10)
   10 FORMAT ('INPUT NSTART, NSTOP,LF, M, IFJ, RFC')
      READ (5,20) NSTART, NSTOP, LF, M, IFJ, RFC
   20 FORMAT (5I5, F5.2)
C
C      LOOP OVER TRACES
C
      ITNUM=NSTOP-NSTART+1
      IF(ITNUM.LT.LF) WRITE(6,21)
   21 FORMAT(/'FILTER LENGTH IS GREATER THAN AUTOCORRELATION WINDOW')
      DO 60 IDEC = 1, N
        WRITE (6,70) ISTAT(IDEC,1) ,ITNUM,LF
        TNUM = 0
        NUM = LX(IDEC)
        IF (NUM .GE. 2048) WRITE (6,80)
        IF (NUM .GE. 2048) STOP
C
C       LOOP OVER SAMPLES
C
        DO 50 ITER = 1, NUMDEC
          TNUM=0
          DO 30 JDEC = 1, NUM
            IF (JDEC .GE. NSTART .AND. JDEC .LE. NSTOP) TNUM = TNUM + 1
            IF (JDEC .GE. NSTART .AND. JDEC .LE. NSTOP) BUFFER(TNUM) =
     1      X(IDEC,JDEC)
   30     TRACE(JDEC) = X(IDEC,JDEC)
          CALL PDECON(TNUM, BUFFER, M, LF, SPACE1, SPACE2, SPACE3,
     1          FILTER, RFC, IFJ)
          LC = NUM + LF - 1
          CALL FOLD(NUM, TRACE, LF, FILTER, LC, SPACE1)
          DO 40 JDEC = 1, NUM
   40     X(IDEC,JDEC) = SPACE1(JDEC)
   50   CONTINUE
   60 CONTINUE
   70 FORMAT ('DECONVOLVING TRACE ', A4,'    AUTO WINDOW=',I5,
     1          '  FILTER LENGTH=',I5)
   80 FORMAT ('TNUM>2048 - CHANGE NSTART,NSTOP')
      RETURN
      END
C
C*********************                          *********************
C
C Subroutine name : PDECON
C
C Description :
C              This subroutine designs a predictive deconvolution
C              filter based on an input time series which may be
C              a source wavelet or a section of a trace. The filter
C              is calculated using the ACF of the input time series
C              and the CCF of the input time series and the desired
C              output, which is merely the input delayed by the
C              prediction distance. Levinson recursion is used to
C              solve the normal equations.
C
C Arguments :
C            LY - Length of input time series for ACF ( > or = LF ).
C             Y - Array containing input time series.
C             M - Prediction distance in samples.
C            LF - Total length of filter. Equal to M + N where
C                 N is order of matrix equation to be solved.
C         R,G,A - Dummy arrays used for internal working.
C             F - Array containing filter coefficients.
C           RFC - Percentage white noise added for stability
C           IFJ - If set to 1, triangular weighting applied to ACF
C
C Subroutines called :
C                      CROSS
C                      LVNSRC
C                      FEJER
C
      SUBROUTINE PDECON(LY, Y, M, LF, R, G, A, F, RFC, IFJ)
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION Y(LY), R(LF), G(LF), A(LF), F(LF)
      N = LF - M
      CALL CROSS(LY, Y, LY, Y, LF, A)
      DO 20 I = 1, LF
        IF (I .GT. N) GO TO 10
        R(I) = A(I)
   10   IF (I .LE. M) GO TO 20
        G(I - M) = A(I)
   20 CONTINUE
      IF (IFJ .EQ. 1) CALL FEJER(R, N)
      R(1) = R(1) * (1.0D0+RFC)
      CALL LVNSRC(N, A, F, R, G)
      F(1) = 1.0D0
      DO 30 I = 2, LF
        IF (I .LE. M) F(I) = 0.0D0
   30 IF (I .GT. M) F(I) = -A(I - M)
      RETURN
      END
C*********************                          *********************
C                              CONVOLUTION
C*********************                          *********************
C
C Subroutine name : FOLD
C
C Description :
C              This subroutine performs polynomial multiplication,
C              or equivalently convolution, of two input time series.
C              The operation is conducted in the time domain.
C
C Arguments :
C           LA - Length in samples of time series A.
C           LB - Length in samples of time series B.
```

```
C         A,B - Arrays containing input time series.
C          LC - Length of output time series, needed for
C               dimensioning purposes. (Equal to LA+LB-1)
C           C - Array containing output time series.
C
C Subroutines called :   ZERO
C
      SUBROUTINE FOLD(LA, A, LB, B, LC, C)
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION A(LA), B(LB), C(LC)
      CALL ZERO(LC, C)
      DO 10 I = 1, LA
        DO 10 J = 1, LB
          K = I + J - 1
   10 C(K) = C(K) + A(I) * B(J)
      RETURN
      END
C
C
C
C********************** SUBROUTINE TTREAD ************************
C
C
C       This routie reads the travel-times held in the travel-time
C       matrices (GPT9:TMAT-[PG, PCP, PMP, PN])
C

      SUBROUTINE TTREAD(X, T, NUM, IQ, ISEA, ISTAT, ISHOT, XCDS)
      INTEGER ASTAT, ASHOT, AMAIN
      DIMENSION T(2000), X(2000), TITLES(20), TITLE(20), ASTAT(110),
     1          ASHOT(110), IST(110), ISH(110), KSTH(180), DIST25(180),
     2          AMAIN(60), ET(110,110), XMAT(110,110), TTN(110,110),
     3          TTM(110,110), TTC(110,110), TTG(110,110), TTN1(110,110)

      DATA NSEA /'N   '/
      DATA BLANK /'    '/
      DATA AMAIN /'1   ', '2   ', '3   ', '4   ', '5   ', '6   ',
     1     '7   ', '8   ', '9   ', '10  ', '11  ', '12  ', '13  ',
     2     '14  ', '15  ', '16  ', '17  ', '18  ', '19  ', '20  ',
     3     '21  ', '22  ', '23  ', '24  ', '25  ', '26  ', '27  ',
     4     '28  ', '29  ', '30  ', '31  ', '32  ', '33  ', '34  ',
     5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
     6     '42  ', '43  ', '44  ', '45  ', '46  ', '47  ', '48  ',
     7     '49  ', '50  ', '51  ', '52  ', '53  ', '54  ', '55  ',
     8     '56  ', '57  ', '58  ', '59  ', '60  '/
C
C
C********************** ASSIGN FILES ******************************
C
C
      CALL FTNCMD('ASSIGN 21=GPT9:TMAT-PN', 22)
      CALL FTNCMD('ASSIGN 29=GPT9:TMAT-PN1', 23)
      CALL FTNCMD('ASSIGN 22=GPT9:TMAT-PMP', 23)
      CALL FTNCMD('ASSIGN 23=GPT9:TMAT-PCP', 23)
      CALL FTNCMD('ASSIGN 24=GPT9:TMAT-PG', 22)
      CALL FTNCMD('ASSIGN 28=GPT9:XMAT-CSSP', 24)
      CALL FTNCMD('ASSIGN 20=GPT9:XMAT-M25B', 24)
C
C      READ HEADER BLOCKS
C
      READ (28,10) TITLES
      READ (28,20) (ASTAT(I),I=1,110)
      READ (28,20) (ASHOT(I),I=1,71)
   10 FORMAT (20A4)
   20 FORMAT (110A4)


C
C
C ********** FIND GIVEN STATION / SHOT IN DATA FILES
C
C
   30 NSH = 0
      NST = 0
      IF (XCDS .NE. 0.0) GO TO 70
      DO 50 I = 1, 110
        IF (ISTAT .EQ. 0 .AND. ISHOT .EQ. 0) GO TO 60
        IF (ISTAT .EQ. 0) GO TO 40
        IF (ISTAT .EQ. ASTAT(I)) NST = I
   40   IF (ISHOT .EQ. 0) GO TO 50
        IF (I .GT. 71) GO TO 50
        IF (ISHOT .EQ. ASHOT(I)) NSH = I
   50 CONTINUE
      ISTAT = NST
      ISHOT = NSH
C
C      COMMON SHOT SECTION WANT TO PLOT ONLY MAINLINE STATION TTS
C
   60 MAINL = 0
      IF (IQ .EQ. 2) MAINL = 1


C
C
C ************************** READ MATRIX FILES
C ************************
C
C
   70 WRITE (6,80)
   80 FORMAT (/'TT BEING READ IN FROM MATRIX FILES')
C
      READ (21,90) T1, S1
      READ (22,90) T2, S2
      READ (23,90) T3, S3
      READ (24,90) T4, S4
      READ (29,90) T4, S4
C
      READ (21,90) T1, S1
      READ (22,90) T2, S2
      READ (23,90) T3, S3
```

```
      READ (24,90) T4, S4
      READ (29,90) T4, S4
C
      READ (21,90) T1, S1
      READ (22,90) T2, S2
      READ (23,90) T3, S3
      READ (24,90) T4, S4
      READ (29,90) T4, S4
C
   90 FORMAT (2A4)
  100 FORMAT (I1)
      M = 0
  110 DO 160 I = 1, 110
        NSHOT = 71
        IF (MAINL .EQ. 0) GO TO 150
        IF (M .LT. 60) GO TO 120
C
C        ONLY 60 STATIONS NEEDED FOR MAINLINE SO FILL ASTAT UP
C        WITH DUMMY VALUES UP TO 110
C
        M = M + 1
        ASTAT(M) = NSEA
        GO TO 160
  120   L = 0
        DO 130 K = 1, 60
          IF (ASTAT(I) .EQ. AMAIN(K)) L = K
  130   CONTINUE
        IF (L .EQ. 0) GO TO 140
        NSHOT = 66
        GO TO 150
  140   READ (28,90) T1, S1
        READ (24,90) T1, S1
        READ (23,90) T1, S1
        READ (22,90) T1, S1
        READ (21,90) T1, S1
        READ (29,90) T1, S1
        GO TO 160
  150   M = M + 1
        IF (ASTAT(I) .EQ. BLANK) GO TO 160
        ASTAT(M) = ASTAT(I)
        IF (ASTAT(ISTAT) .EQ. ASTAT(M)) ISTAT = M
        READ (28,180) (XMAT(M,J),J=1,NSHOT)
        READ (24,170) (TTG(M,J),ET(M,J),J=1,NSHOT)
        READ (23,170) (TTC(M,J),ET(M,J),J=1,NSHOT)
        READ (22,170) (TTM(M,J),ET(M,J),J=1,NSHOT)
        READ (21,170) (TTN(M,J),ET(M,J),J=1,NSHOT)
        READ (29,170) (TTN1(M,J),ET(M,J),J=1,NSHOT)
  160 CONTINUE
  170 FORMAT (71(2F10.4))
  180 FORMAT (71F10 4)
      READ (20,10) TITLES
      READ (20,190) (KSTH(I),I=1,160)
      READ (20,200) (DIST25(I),I=1,160)
  190 FORMAT (160A4)
  200 FORMAT (160F10.4)
  210 IF (IQ .EQ. 1) GO TO 220
      IF (IQ .EQ. 2) GO TO 300
      IF (IQ .EQ. 3) GO TO 480
      IF (IQ .EQ. 4) GO TO 370
C
C
C        COMMON STATION
C
C
  220 IF (ISEA .EQ. NSEA) GO TO 230
      NSTRT = 1
      NEND = NSTRT + 36 - 1
      GO TO 240
  230 NSTRT = 37
      NEND = NSTRT + 30 - 1
  240 L = 0
      DO 290 I = NSTRT, NEND
        IST(L) = ISTAT
        ISH(L) = I
        IF (TTN1(ISTAT,I) .EQ. 0.0) GO TO 250
        L = L + 1
        T(L) = TTN1(ISTAT,I)
        X(L) = XMAT(ISTAT,I)
  250   IF (TTN(ISTAT,I) .EQ. 0.0) GO TO 260
        L = L + 1
        T(L) = TTN(ISTAT,I)
        X(L) = XMAT(ISTAT,I)
  260   IF (TTM(ISTAT,I) .EQ. 0.0) GO TO 270
        L = L + 1
        T(L) = TTM(ISTAT,I)
        X(L) = XMAT(ISTAT,I)
  270   IF (TTC(ISTAT,I) .EQ. 0.0) GO TO 280
        L = L + 1
        T(L) = TTC(ISTAT,I)
        X(L) = XMAT(ISTAT,I)
  280   IF (TTG(ISTAT,I) .EQ. 0.0) GO TO 290
        L = L + 1
        T(L) = TTG(ISTAT,I)
        X(L) = XMAT(ISTAT,I)
  290 CONTINUE
      NUM = L
      RETURN
C
C
C        COMMON SHOT
C
C
  300 L = 0
      DO 360 I = 1, 60
C
        IF (TTN1(I,ISHOT) .EQ. 0.0) GO TO 320
        L = L + 1
        T(L) = TTN1(I,ISHOT)
        X(L) = XMAT(I,ISHOT)
        WRITE (7,310) ASTAT(I), ASHOT(ISHOT)
```

```
 310   FORMAT (2A4)
 320   IF (TTN(I,ISHOT) .EQ. 0.0) GO TO 330
       L = L + 1
       X(L) = XMAT(I,ISHOT)
       T(L) = TTN(I,ISHOT)
 330   IF (TTM(I,ISHOT) .EQ. 0.0) GO TO 340
       L = L + 1
       X(L) = XMAT(I,ISHOT)
       T(L) = TTM(I,ISHOT)
 340   IF (TTC(I,ISHOT) .EQ. 0.0) GO TO 350
       L = L + 1
       X(L) = XMAT(I,ISHOT)
       T(L) = TTC(I,ISHOT)
 350   IF (TTG(I,ISHOT) .EQ. 0.0) GO TO 360
       L = L + 1
       X(L) = XMAT(I,ISHOT)
       T(L) = TTG(I,ISHOT)
 360 CONTINUE
     NUM = L
     RETURN
C
C
C     COMMON MIDPOINT
C
C
 370 DO 380 I = 1, 180
       IF (ASTAT(ISTAT) .EQ. KSTH(I)) NST = I
       IF (ASHOT(ISHOT) .EQ. KSTH(I)) NSH = I
 380 CONTINUE
     X1 = (DIST25(NST) + DIST25(NSH)) / 2.0
 390 XU = X1 + 0.5
     XD = X1 - 0.5
     LL = 0
     DO 470 I = 1, 180
       NSTRT = I + 1
       IF (I .EQ. 180) NSTRT = NSTRT - 1
       DO 470 J = NSTRT, 180
         XMP = (DIST25(I) + DIST25(J)) / 2.0
         IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 470
         LL = LL + 1
         IST(LL) = 0
         ISH(LL) = 0
         NEND = 110
         IF (MAINL .EQ. 1) NEND = 60
         DO 410 IJ = 1, NEND
           IF (KSTH(I) .EQ. ASTAT(IJ)) IST(LL) = IJ
           IF (KSTH(J) .EQ. ASTAT(IJ)) IST(LL) = IJ
 400       IF (IJ .GT. 71) GO TO 410
           IF (KSTH(J) .EQ. ASHOT(IJ)) ISH(LL) = IJ
           IF (KSTH(I) .EQ. ASHOT(IJ)) ISH(LL) = IJ
 410     CONTINUE
C
C         IF IST(L) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C         RESET L AND IST AND ISH
C
         IF (IST(LL) .EQ. 0 .OR. ISH(LL) .EQ. 0) GO TO 460
         M = IST(LL)
         K = ISH(LL)
         IF (TTN1(M,K) .EQ. 0.0) GO TO 420
         L = L + 1
         T(L) = TTN1(M,K)
         X(L) = XMAT(M,K)
 420     IF (TTN(M,K) .EQ. 0.0) GO TO 430
         L = L + 1
         T(L) = TTN(M,K)
         X(L) = XMAT(M,K)
 430     IF (TTM(M,K) .EQ. 0.0) GO TO 440
         L = L + 1
         X(L) = XMAT(M,K)
         T(L) = TTM(M,K)
 440     IF (TTC(M,K) .EQ. 0.0) GO TO 450
         L = L + 1
         T(L) = TTC(M,K)
         X(L) = XMAT(M,K)
 450     IF (TTC(M,K) .EQ. 0.0) GO TO 460
         L = L + 1
         T(L) = TTG(M,K)
         X(L) = XMAT(M,K)
         GO TO 470
 460     IST(LL) = 0
         ISH(LL) = 0
         LL = LL - 1
 470 CONTINUE
     NUM = L
     RETURN
C
C
C       COMMON DISTANCE
C
C
 480 XU =XCDS + 1
     XD =XCDS - 1
     LL = 0
     NEND = 110
     NEND1 = 71
     IF (MAINL .EQ. 1) NEND = 60
     IF (MAINL .EQ. 1) NEND1 = 66
     DO 490 I = 1, NEND
       DO 490 J = 1, NEND1
         IF (XMAT(I,J) .GT. XU .OR. XMAT(I,J) .LT. XD) GO TO 490
         LL = LL + 1
         IST(LL) = I
         ISH(LL) = J
 490 CONTINUE
     NO = LL
     L = 0
     DO 540 I = 1, NO
       M = IST(I)
       K = ISH(I)
       IF (TTN1(M,K) .EQ. 0.0) GO TO 500
```

```
      L = L + 1
      T(L) = TTN1(M,K)
      X(L) = XMAT(M,K)
 500  IF (TTN(M,K) .EQ. 0.0) GO TO 510
      L = L + 1
      T(L) = TTN(M,K)
      X(L) = XMAT(M,K)
 510  IF (TTM(M,K) .EQ. 0.0) GO TO 520
      L = L + 1
      X(L) = XMAT(M,K)
      T(L) = TTM(M,K)
 520  IF (TTC(M,K) .EQ. 0.0) GO TO 530
      L = L + 1
      T(L) = TTC(M,K)
      X(L) = XMAT(M,K)
 530  IF (TTG(M,K) .EQ. 0.0) GO TO 540
      L = L + 1
      T(L) = TTG(M,K)
      X(L) = XMAT(M,K)
 540 CONTINUE
     NUM = L
C
CCCCCCCCCCCCCCC
C
C     IF COMMON DISTANCE PLOT REQUIRED NEED TO CHANGE DISTANCES
C     TO DIST FROM M25B OF EACH PARTICULAR STAT/SHOT'S  MID-POINT
C
CCCCCCCCCCCCCCC
C
     DO 560 I = 1, NUM
       M = IST(I)
       K = ISH(I)
       DO 550 J = 1, 160
         IF (ASTAT(M) .EQ. KSTH(J)) NST = J
         IF (ASHOT(K) .EQ. KSTH(J)) NSH = J
 550   CONTINUE
       X(I) = (DIST25(NST) + DIST25(NSH)) / 2.0
 560 CONTINUE
     RETURN
     END
C
```

```
C
CA    AHJ Lewis
CD+
C*****************************************************************
C*                                                               *
C*                                                               *
C*                    PROGRAMME TMATCOPY.U                       *
C*                                                               *
C*                                                               *
C* AHJ LEWIS      Dept. Geol. Sciences, Durham          Nov,1985 *
C*****************************************************************
C
C                        DESCRIPTION
C
C
C     The programme copies travel-times to the travel-time matrices.
C     The travel-times can be inputted along CSH, CST, OMP or CDS
C     lines.
C     The programme will calculate travel-times from measurements
C     on paper playouts; will correct travel-times for corrections
C     measured on paper playouts and will accept raw travel-times.
C     The facility to read output from the PERQ digitizer and
C     calculate travel-times is also available.
C
C     NOTES   1) To input 0 tt to a location type in 999.9 - a zero
C     -----        is used to represent a blank location with no tt.
C             2) Checks are made that the correct matix has been assigned.
C
C
C
C
C     INPUT
C     -----
C     The programme is for interactive use and prompts the user for any
C     output required. Units are assigned as follows:-
C
C     unit 3 - file containing PERQ digitizer output
C     unit 4 - travel-time matrix file for phase P
C     unit 5 - default
C     unit 6 - default
C     unit 7 - list of times inputted to matrix
C     unit 8 - assigned internally to GPT9:XMAT-CSSP.
C              GPT9:XMAT-CSSP is the CSSP distance matrix
C     unit 10- assigned internally to GPT9:XMAT-M25B.
C              GPT9:XMAT-CSSP holds the dist from M25B of all CSSP sites.
C
C_________________________________________________________________
CD-
```

```
      INTEGER ASHOT, ASTAT, XSTAT, XSHOT, X1, X2, X3, X4, Y1, Y2, Y3,
     .        Y4, KIND
      DIMENSION TT(110,110), ETT(110,110), ASHOT(110), ASTAT(110),
     .          XMAT(110,110), KSTH(180), DIST25(180), IST(110),
```

```
     .          ISH(110), XSTAT(110), XSHOT(110)
      COMMON X1, Y1, X2, Y2


C
C---- ASSIGN DATA FILES TO UNIT NUMBERS AND READ DATA
C

      DATA NSEA /'N   '/
      DATA PG /'PG  '/
      DATA PC1 /'PC1 '/
      DATA PC2 /'PC2 '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PM1P /'PM1P'/
      DATA PM2P /'PM2P'/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
      DATA M25A /'M25A'/
      DATA M25B /'M25B'/


C
C------------ FIND OUT WHICH DATA IS TO BE INPUT
C
C     MEM=0 - first time through programme so read all matrices
C     MEM=1 - read only new travel-time matix on unit 4

      MEM = 0


C output to screen unit associations

      WRITE (6,10)
   10 FORMAT (/'UNIT SPECIFICATION FOR MATCOPY IS :-'/
     .        'UNIT 3  = DIGITISED TT FILE'//
     .        'ASSIGNED INTERNALLY:-'/
     .        'UNIT 4  = GPT9:TMAT-PHASE'/
     .        'UNIT 5  = *SOURCE*'/
     .        'UNIT 6  = *SOURCE'/
     .        'UNIT 7  = LIST OF TIMES INPUTTED'/
     .        'UNIT 8  = GPT9:XMAT-CSSP'/
     .        'UNIT 10 = GPT9:XMAT-M25B')


C read which phase required and assign files to units

   20 WRITE (6,30)
   30 FORMAT (/'TRAVEL TIMES FOR WHICH PHASE? (EG. PG,PC1,PCP,PMP,PN)')
      READ (5,40) P
   40 FORMAT (A4)
      WRITE (7,40) P
      IF (P .EQ. PG) GO TO 50
      IF (P .EQ. PC1) GO TO 60
      IF (P .EQ. PC2) GO TO 70
      IF (P .EQ. PCP) GO TO 80
      IF (P .EQ. PMP) GO TO 90
      IF (P .EQ. PM1P) GO TO 100
      IF (P .EQ. PM2P) GO TO 110
      IF (P .EQ. PN) GO TO 120
      IF (P .EQ. PN1) GO TO 130
   50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 140
   60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC1', 22)
      GO TO 140
   70 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC2', 22)
      GO TO 140
   80 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 140
   90 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 140
  100 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM1P', 23)
      GO TO 140
  110 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM2P', 23)
      GO TO 140
  120 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 140
  130 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 140
  140 CONTINUE
      IF (MEM .EQ. 1) GO TO 150
      CALL FTNCMD('ASSIGN 8=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 1=GPT9:XMAT-M25B', 23)


C check correct matrix file is assigned to unit 4

  150 READ (4,510) TIN
      IF (TIN .EQ. P) GO TO 170
      WRITE (6,160) TIN, P
  160 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
      STOP


C read values from matrix file

  170 WRITE (6,180)
  180 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (4,190) (ASTAT(I),I=1,110)
      READ (4,190) (ASHOT(I),I=1,71)
      IF (MEM .EQ. 1) GO TO 200
      READ (8,510) ADUMP
      READ (8,190) (XSTAT(I),I=1,110)
      READ (8,190) (XSHOT(I),I=1,71)
  190 FORMAT (110A4)
  200 DO 240 I = 1, 110
        READ (4,210) (TT(I,J),ETT(I,J),J=1,71)
```

```
      IF (MEM .EQ. 1) GO TO 220
      READ (8,210) (XMAT(I,J),J=1,71)
 210  FORMAT (71(F10.4,F10.4))
 220  IF (ASTAT(I) .EQ. XSTAT(I) .AND. ASHOT(I) .EQ. XSHOT(I))
    1     GO TO 240
      WRITE (6,230)
 230  FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
      STOP
 240 CONTINUE

C values read



C
C       NOW FIND OUT WHAT FORM DATA IS IN
C
C

 250 WRITE (6,260)
 260 FORMAT (/'COMMON STATION =1 OR COMMON SHOT =2 OR CODS & ENDS= 3 OR
    1'/'COMMON MID-POINT =4 OR COMMON DISTANCE =5 ? '/'_')
     READ (5,270) KIND
 270 FORMAT (I1)
     WRITE (6,280)
 280 FORMAT (/'IF TRAVEL TIMES TO BE INPUT TYPE            0'/
    1        'IF TT TO BE CALCULA TED TYPE                1'/
    2        'IF DIGITIZED TT TO BE INPUT FROM UNIT3 TYPE 2'/
    3        'IF CORRECTION TO TT (+/-) TO BE INPUT TYPE  3')
     READ (5,270) IQTT


C ICORR=0 - set for routine which read the digitizer output on unit3
C IQTT    - switch for type of data to be input

     ICORR=0
     IF (IQTT .EQ. 0) GO TO 400
     IF (IQTT .EQ. 1) GO TO 340
     IF (IQTT .EQ. 3) GO TO 370



C
C IQTT=2 - read tt data from unit3 . the PERO digitizer data
C

     READ (3,290) ISTYPE, VRED, TSTRT, TOTT
 290 FORMAT (I1, 3F5.2)
     IF (ISTYPE .EQ. KIND) GO TO 310
     WRITE (6,300)
 300 FORMAT (/'DIGITIZED FILE INCORRECT!')
     STOP
```

```
C read in station shot or cds and which repeated hot is used (A or B)
C read in corner positions of plot from unit3

 310 READ (3,320) LSTAT, LSHOT, CDSL, N23
 320 FORMAT (2A4, 2X, F10.4, A4)
     READ (3,330) ITYPE, X1, Y1
     READ (3,330) ITYPE, X2, Y2
     READ (3,330) ITYPE, X3, Y3
     READ (3,330) ITYPE, X4, Y4
 330 FORMAT (I1, 2I6)

C calculate scale of plot in digitized units

     DTOT1 = ((Y4 - Y1)**2 + (X4 - X1)**2) ** 0.5
     DTOT2 = ((Y3 - Y2)**2 + (X3 - X2)**2) ** 0.5
     DTOT = (DTOT1 + DTOT2) / 2.0
     DIGSCL = TOTT / DTOT

C completed go to 400

     GO TO 400



C
C IQTT=1 - read in plot maesurements. TT measured off paper plots
C

 340 WRITE (6,350)
 350 FORMAT (/'INPUT START TIME, SCALE (MM/SEC), RED. VEL. OF SECTION'/
    1        '_____ _____ _____ 3F5.2')
     READ (5,360) TSTRT, TSCALE, VRED
 360 FORMAT (3F5.2)
     GO TO 400



C
C IQTT=3 - correction to tt's to be input in distance units
C


 370 WRITE (6,380)
 380 FORMAT (/'INPUT PLOT SCALE F5.2')
     READ (5,390) TSCALE
 390 FORMAT (F5.2)
     ICORR = IQTT
     IQTT = 0
     GO TO 400
```

```
C
C     FIND OUT WHICH TYPE OF PLOT USED TO MEASURE TTRAVEL-TIMES
C
C

 400 WRITE (6,410)
 410 FORMAT (/'TO TERMINATE DATA INPUT AT ANY TIME TYPE IN -VE TIME'/
    1        'TO INPUT A ZERO TT DO NOT TYPE TT AS 0.0 BUT AS 999.9')
     IF (KIND .EQ. 1) GO TO 420
     IF (KIND .EQ. 2) GO TO 660
     IF (KIND .EQ. 3) GO TO 830
     IF (KIND .EQ. 4) GO TO 1000
     IF (KIND .EQ. 5) GO TO 1160



C
C*****************************************************************
C*                                                               *
C*              COMMON STATION : KIND=1                          *
C*                                                               *
C*****************************************************************
C
C
C
 420 WRITE (6,430)
 430 FORMAT (/'WHICH STATION A4 ?'/'____')
     READ (5,440) NSTAT
 440 FORMAT (A4)


C IQTT=2 check correct station is specified in unit3

     IF (LSTAT .EQ. NSTAT .OR. IQTT .NE. 2) GO TO 460
     WRITE (6,300)
     WRITE (6,450) LSTAT, NSTAT, IQTT
 450 FORMAT ('DIDNOT GET TO DIGIN', 2A4, I1)
     STOP

C find station in matrix

 460 NST = 0
     DO 470 I = 1, 110
 470 IF (NSTAT .EQ. ASTAT(I)) NST = I
     IF (NST .NE. 0) GO TO 490
     WRITE (6,480) NSTAT
 480 FORMAT (/'STATION  NOT FOUND IN MATRIX ', A4)
     GO TO 1290

C IF STATION FOUND - CONTINUE

 490 NSTAT = NST
     WRITE (6,500)
```

```
 500 FORMAT (/'NORTH OR IRISH SEA? (N OR M)'/'_')
     READ (5,510) ISEA
 510 FORMAT (A4)
     IF (ISEA .EQ. NSEA) GO TO 520
     NSTRT = 1
     NEND = NSTRT + 36 - 1
     GO TO 530
 520 NSTRT = 37
     NEND = NSTRT + 30 - 1


C
C LOOP OVER SHOTS : TT ARE INPUTTED DIRECTLY          IQTT =0
C                   TT CORRECTED IF ICORR=3           IQTT =3
C                   CALCULATED FROM DISTANCES         IQTT =1
C                   CALCULATED FROM DIGIT FILE        IQTT =2
C                   CHECKS WHICH REPEATED SHOT
C                   WAS PLOTTED ON SECTION
C

 530 DO 650 I = NSTRT, NEND
       IF (ASHOT(I) .EQ. N23A .AND. N23 .EQ. N23B) GO TO 650
       IF (ASHOT(I) .EQ. N23B .AND. N23 .EQ. N23A) GO TO 650
       IF (ASHOT(I) .EQ. M25A .AND. N23 .EQ. M25B) GO TO 650
       IF (ASHOT(I) .EQ. M25B .AND. N23 .EQ. M25A) GO TO 650
       IF (IQTT .EQ. 0) GO TO 580
       IF (IQTT .EQ. 1) GO TO 560

C IQTT =2 : CALCULATE TT FROM UNIT 3

       CALL DIGIN(DIGSCL, T, ET)

C IF BLANK LINE READ IN UNIT 3 RETURNS T<0.0
C IF T<0.0 GO TO END OF TT CALCULATION AND INPUT SECTION
C IF NOT CONTINUE IN LOOP
C T=0.0 IF NO TT PICK MADE AT PRESENT SHOT

       IF (T .GE. 0.0) GO TO 550
       INUM = I - NSTRT
       WRITE (6,540) INUM
 540   FORMAT (/'END OF DIGITIZED FILE REACHED AFTER ', I5, ' TRACES')
       GO TO 1290
 550   IF (T .EQ. 0.0) GO TO 600
       T = T + TSTRT + XMAT(NSTAT,I) / VRED
       GO TO 600
C
C IQTT =1 : READ IN DISTANCES & USE TO CALCULATE TT
C

 560   WRITE (6,1280) P, ASTAT(NSTAT), NSTAT, ASHOT(I), I
       READ (5,590) T, ET
       IF (T .LT. 0.0) GO TO 1290
       IF (T .EQ. 0.0) GO TO 650
       IF (T .EQ. 999.9 .AND. TT(NSTAT,I) .EQ. 0.0) GO TO 650
```

```
      IF (T .EQ. 999.9) GO TO 570
      T = (T/TSCALE) + TSTRT + XMAT(NSTAT,I) / VRED
      ET = ET / TSCALE
 570  GO TO 600


C IQTT =0 : READ IN TT AND ERROR

 580  WRITE (6,1270) P, ASTAT(NSTAT), NSTAT, ASHOT(I), I
      READ (5,590) T, ET
 590  FORMAT (2F10.4)
      IF (T .EQ. 999.9) GO TO 600
      IF (T .LT. 0.0.AND.ICORR.NE.3) GO TO 1290
      IF (ICORR .EQ. 3) T = TT(NSTAT,I) + T / TSCALE
      IF (ICORR .EQ. 3) ET = ETT(NSTAT,I)


C TT NOW CALCULATED :
C INPUT THESE INTO ARRAYS BUT FIRST
C COMPARE TO ORIGINAL VALUES IN MATRICES
C IF DIFFERENT GIVE OPTIONS OF IGNORING
C THE NEW OR OVERWRITING THE OLD VALUES

C IF T=0.0              -NO PICK MADE          -GO TO NEXT SHOT
C IF TT(NSTAT,I)=0.0    -ORIGINAL TT=0.0       -COPY NEW TT
C IF TT(NSTAT,I)=T      -ORIGINAL TT=NEW TT    -GO TO NEXT SHOT
C IF T=999.9            -SET TT=0.0            -GIVE OPTION TO
C                                               COPY NEW TT
C
C IF PASSES THROUGH ALL IF'S GIVE OPTION TO COPY NEW TT AS
C T NOT EQUAL TO TT(NSTAT,I) AND SO NEW IS NOT EQUAL TO OLD TT.


 600  IF (T .EQ. 0) GO TO 650
      IF (T .LT. 0.0) GO TO 1290
      IF (TT(NSTAT,I) .EQ. 0.0 .AND. T .NE. 999.9) GO TO 630
      IF (T .EQ. TT(NSTAT,I)) GO TO 650
      IF (T .EQ. 999.9) T = 0.0
      IF (T .EQ. 0.0) ET = 0.0
      WRITE (6,610) ASHOT(I), TT(NSTAT,I), T
 610  FORMAT (/'PREVIOUS TT AND VALUE JUST INPUT DISAGREE FOR SHOT',
     1        A4, /'CONT WITH NEXT SHOT: OVERWRITE OLD TT ? Y=1,N=0'/
     2        'OLD & NEW TT ARE RESPECTIVELY ', 2F10.4, /'_')
      READ (5,270) IQ
      IF (IQ .EQ. 1) GO TO 630
      WRITE (6,620)
 620  FORMAT (/'CONTINUE WITH NEXT SHOT BUT LEAVE OLD TT? Y=1 N=0'/
     1        '_')
      READ (5,270) IQ
      IF (IQ .EQ. 1) GO TO 650

C IF IQ=0 JUMP OUT OF LOOP - DO NOT CONTINUE WITH NEXT SHOT

      GO TO 1290

C
C                 ____________________
C                 COPY NEW TT INTO ARRAY
C                 ____________________

 630  TT(NSTAT,I) = T
      ETT(NSTAT,I) = ET
      WRITE (7,640) ASTAT(NSTAT), ASHOT(I), T, ET
 640  FORMAT (/'TT FOR ', A4, ' AND ', A4, ' INPUT IS ', 2F10.3)

C LOOP TO NEXT SHOT

 650 CONTINUE


C ALL SHOTS DONE GO TO END OF TT INPUT SECTION

     GO TO 1290

C
C*************** END TO COMMON STATION SECTION *******************
C




C
C*****************************************************************
C*                                                               *
C*    COMMON SHOT     : KIND=2                                   *
C*                                                               *
C*****************************************************************
C
 660 WRITE (6,670)
 670 FORMAT (/'WHICH SHOT (A4) ?'/'____')
     READ (5,680) NSHOT
 680 FORMAT (A4)


C     FIND SHOT'S POSITION IN MATRICES
C     IF SHOT FOUND CONTINUES
C     IF SHOT NOT FOUND GOES TO END OF INPUT SECTIONS


 690 NSH = 0
     DO 700 I = 1, 71
 700 IF (NSHOT .EQ. ASHOT(I)) NSH = I
     IF (NSH .NE. 0) GO TO 720
     WRITE (6,710) NSHOT
 710 FORMAT (/'SHOT ', A4, ' NOT FOUND')
     GO TO 1290


C
C     LOOP OVER STATS : TT ARE INPUTTED DIRECTLY    IQTT =0
```

```
C                          CORRECTION TO TT ICORR=3 &  IOTT =0
C                          CALCULATED FROM DISTANCES   IOTT =1
C                          CALCULATED FROM DIGIT FILE  IOTT =2

 720 NSHOT = NSH
     DO 810 I = 1, 110
       IF (IOTT .EQ. 0) GO TO 750
       IF (IOTT .EQ. 1) GO TO 730

C IOTT =2 : CALCULATE TT FROM UNIT 3

       CALL DIGIN(DIGSCL, T, ET)


C       IF BLANK LINE READ IN UNIT 3 RETURNS T<0.0
C       IF T<0.0 GO TO END OF TT CALCULATION AND INPUT SECTION
C       IF NOT CONTINUE IN LOOP
C       T=0.0 IF NO TT PICK MADE AT PRESENT SHOT

       IF (T .LT. 0.0) GO TO 1290
       IF (T .EQ. 0.0) GO TO 810
       T = T + TSTRT + XMAT(I,NSHOT) / VRED
       GO TO 770

C IOTT =1 : READ IN DISTANCES AND USE TO CALCULATE TT

 730   WRITE (6,1280) P, ASTAT(I), I, ASHOT(NSHOT), NSHOT
       READ (5,590) T, ET
       IF (T .LT. 0.0) GO TO 1290
       IF (T .EQ. 0.0) GO TO 810
       IF (T .EQ. 999.9 .AND. TT(I,NSHOT) .EQ. 0.0) GO TO 810
       IF (T .EQ. 999.9) GO TO 740
       T = (T/TSCALE) + TSTRT + XMAT(I,NSHOT) / VRED
       ET = ET / TSCALE
 740   GO TO 770

C IOTT =0 : READ IN TT AND ERROR

 750   WRITE (6,1270) P, ASTAT(I), I, ASHOT(NSHOT), NSHOT
       READ (5,760) T, ET
 760   FORMAT (2F10.4)
       IF (T .EQ. 999.9) GO TO 770
       IF (T .LT. 0.0.AND.ICORR.NE.3) GO TO 1290
       IF (ICORR .EQ. 3) T = TT(I,NSHOT) + T / TSCALE
       IF (ICORR .EQ. 3) ET = ETT(I,NSHOT)


C       TT NOW CALCULATED
C       INPUT THESE INTO ARRAYS BUT FIRST
C       COMPARE TO ORIGINAL VALUES IN MATRICES
C       IF DIFFERENT GIVE OPTIONS OF IGNORING
C       THE NEW OR OVERWRITING THE OLD VALUES
C
C       IF T=0.0 NO PICK MADE AT THIS SHOT          -GO TO NEXT SHOT
C       IF TT(I,NSHOT)=0.0  -ORIGINAL TT=0.0        -COPY NEW TT
C       IF TT(I,NSHOT)=T    -ORIGINAL TT= NEW TT    -GO TO NEXT SHOT
C       IF T=999.9          -SET TT=0.0             -GIVE OPTION TO
C                                                    COPY NEW TT
C       IF PASSES THROUGH ALL IF'S GIVE OPTION TO COPY NEW TT AS
C       T NOT EQUAL TO TT(NSTAT,I) AND SO NEW IS NOT EQUAL TO OLD TT.


 770   IF(T.EQ.0.0) GOTO 810
       IF (T .LT. 0.0) GO TO 1290
       IF (TT(I,NSHOT) .EQ. 0.0.AND.T.NE.999.9) GO TO 800
       IF (T .EQ. TT(I,NSHOT)) GO TO 810
       IF (T .EQ. 999.9) T = 0.0
       IF (T .EQ. 999.9) ET = 0.0
       WRITE (6,780) ASTAT(I), TT(I,NSHOT), T
 780   FORMAT (/'PREVIOUS TT AND VALUE JUST INPUT DISAGREE FOR STAT',
     1         A4, /'CONT WITH NEXT STAT: OVERWRITE OLD TT ? Y=1,N=0'/
     2         'OLD & NEW TT ARE RESPECTIVELY ', 2F10.4, /'_')
       READ (5,270) IQ
       IF (IQ .EQ. 1) GO TO 800
       WRITE (6,790)
 790   FORMAT (/'CONTINUE WITH NEXT SHOT BUT LEAVE OLD TT ? Y=1'/'_')
       READ (5,270) IQ
       IF (IQ .EQ. 1) GO TO 810

C       IF IQ=0 DO NOT GO TO NEXT STATION - JUMP OUT OF LOOP

       GO TO 1290

C
C                    ____________________
C                    COPY NEW TT TO ARRAY
C                    ____________________

 800   TT(I,NSHOT) = T
       ETT(I,NSHOT) = ET
       WRITE (7,640) ASTAT(I), ASHOT(NSHOT), T, ET

C      LOOP TO NEXT STATION

 810 CONTINUE

C      ALL STATIONS DONE - OPTION TO DO ANOTHER COMMON SHOT

      WRITE (6,820)
 820 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER SHOT? YES=1 NO=0',
     1       /'_')
      READ (5,270) IQ
      IF (IQ .EQ. 1) GO TO 660

C      ALL STATIONS DONE GO TO END OF TT INPUT SECTION

      GO TO 1290

C
C*************** END TO COMMON SHOT SECTION ********************
C
```

```
C
C*****************************************************************
C*                                                               *
C*    CODS AND BOBS  : KIND=3                                    *
C*                                                               *
C*****************************************************************
C

 830 WRITE (6,840)
 840 FORMAT (/'WHICH STATION (A4)  & SHOT (A4)'/'____/____')
     READ (5,850) NSTAT, NSHOT
 850 FORMAT (A4, 1X, A4)

C     IQTT=2 READING DIGITISED FILE : IF SO THEN CHECK CORRECT
C                                     SHOT IS SPECIFIED

     IF (IQTT .NE. 2) GO TO 860
     IF (LSTAT .EQ. NSTAT .AND. LSHOT .EQ. NSHOT) GO TO 860
     WRITE (6,300)
     STOP

C     FIND TRACE'S POSITION IN MATRICES
C     IF FOUND CONTINUES
C     IF NOT FOUND GOES TO END OF INPUT SECTIONS

 860 NSH = 0
     NST = 0
     DO 870 I = 1, 110
       IF (NSTAT .EQ. ASTAT(I)) NST = I
       IF (I .GT. 71) GO TO 870
 870 IF (NSHOT .EQ. ASHOT(I)) NSH = I
     IF (NST .NE. 0) GO TO 890
     IF (NSH .NE. 0) GO TO 890
     WRITE (6,880) NSTAT, NSHOT
 880 FORMAT (/'STATION OR SHOT NOT FOUND IN MATRIX ', A4, 1X, A4)
     GO TO 1290
 890 NSTAT = NST
     NSHOT = NSH


C
C     TRACE POSITION GOT-TT ARE INPUTTED DIRECTLY     IQTT =0
C                           CALCULATED FROM DISTANCES   IQTT =1
C                           CALCULATED FROM DIGIT FILE  IQTT =2

     IF (IQTT .EQ. 0) GO TO 920
     IF (IQTT .EQ. 1) GO TO 900

C IQTT =2 : CALCULATE TT FROM UNIT 3

     CALL DIGIN(DIGSCL, T, ET)
```

```
C       IF BLANK LINE READ IN UNIT 3 RETURNS T<0.0
C       IF T<0.0 GO TO END OF TT CALCULATION AND INPUT SECTION
C       IF NOT CONTINUE
C       T=0.0 IF NO TT PICK MADE AT PRESENT SHOT


     IF (T .LE. 0.0) GO TO 940
     T = T + TSTRT + XMAT(NSTAT,NSHOT) / VRED
     GO TO 940

C IQTT =1 : READ IN DISTANCES AND USE TO CALCULATE TT

 900 WRITE (6,1280) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
     READ (5,590) T, ET
     IF (T .LE. 0.0) GO TO 940
     IF (T .EQ. 999.9 .AND. TT(NSTAT,NSHOT) .EQ. 0.0) GO TO 980
     IF (T .EQ. 999.9) GO TO 910
     T = (T/TSCALE) + TSTRT + XMAT(NSTAT,NSHOT) / VRED
     ET = ET / TSCALE
 910 GO TO 940

C       IQTT =0 : READ IN TT AND ERROR

 920 WRITE (6,1270) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
     READ (5,930) T, ET
 930 FORMAT (2F10.4)


C       TT NOW CALCULATED :
C       INPUT THESE INTO ARRAYS BUT FIRST
C       COMPARE TO ORIGINAL VALUES IN MATRICES
C       IF DIFFERENT GIVE OPTIONS OF IGNORING
C       THE NEW OR OVERWRITING THE OLD VALUES
C
C       IF T=0.0 NO PICK MADE AT THIS SHOT        -GO TO NEXT TRACE
C       IF TT(NSTAT,I)=0.0  -ORIGINAL TT=0.0      -COPY NEW TT
C       IF TT(NSTAT,I)=T    -ORIGINAL TT= NEW TT  -GO TO NEXT TRACE
C       IF T=999.9          -SET TT=0.0           -GIVE OPTION TO
C                                                  COPY NEW TT
C       IF PASSES THROUGH ALL IF'S GIVE OPTION TO COPY NEW TT AS
C       T NOT EQUAL TO TT(NSTAT,I) AND SO NEW IS NOT EQUAL TO OLD TT.

 940 IF (T .LT. 0) GO TO 1290
     IF (TT(NSTAT,NSHOT) .EQ. 0.0) GO TO 970
     IF (T .EQ. TT(NSTAT,NSHOT)) GO TO 970
     IF (T .EQ. 999.9) T = 0.0
     IF (T .EQ. 999.9) ET = 0.0
     WRITE (6,950) ASTAT(NSTAT), ASHOT(NSHOT), TT(NSTAT,NSHOT), T
 950 FORMAT (/'PREVIOUS TT AND VALUE JUST INPUT DISAGREE FOR STAT & SHO
    1 T', A4, 1X, A4, /'OVERWRITE OLD TT=', F10.4, ' BY NEW TT=',
    2       F10.3, /'Y=1,N=0'/'_')
     READ (5,270) IQ
     IF (IQ .EQ. 1) GO TO 970
     WRITE (6,960)
```

```
 960 FORMAT (/'CONTINUE WITH NEXT VALUE BUT LEAVE OLD TT? Y=1'/'_')
     READ (5,270) IQ
     IF (IQ .EQ. 1) GO TO 830

C        IF IQ=0 DO NOT GO TO NEXT STATION - JUMP OUT OF LOOP

     GO TO 1290

C                    ____________________
C                    COPY NEW TT TO ARRAY
C                    ____________________

 970 TT(NSTAT,NSHOT) = T
     ETT(NSTAT,NSHOT) = ET
     WRITE (7,640) ASTAT(NSTAT), ASHOT(NSHOT), T, ET

C     TRACE DONE - OPTION TO DO ANOTHER TRACE

 980 WRITE (6,990)
 990 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER VALUE? YES=1 NO=0'/
    1        '_')
     READ (5,270) IQ
     IF (IQ .EQ. 1) GO TO 830

C     GO TO END OF INPUT SECTION IF NO MORE TRACES REQUIRED

     GO TO 1290

C
C*************** END TO CODS AND BOBS SECTION ******************
C



C
C*****************************************************************
C*                                                               *
C*    COMMON MID-POINT : KIND=4                                  *
C*                                                               *
C*****************************************************************
C


1000 WRITE (6,1010)
1010 FORMAT (/'INPUT NEAREST STAT/SHOT PAIR TO DEFINE OMP'/'____/____')
     READ (6,1020) NSTAT, NSHOT
1020 FORMAT (A4, 1X, A4)
     IF (LSTAT .EQ. NSTAT .AND. LSHOT .EQ. NSHOT) GO TO 1030
     WRITE (6,300)
     STOP

C     FIND TRACE'S POSITION IN MATRICES
C     IF FOUND CONTINUES
C     IF NOT FOUND GOES TO END OF INPUT SECTIONS
C
C     ALSO CALCULATE WHICH STHOT/STATION PAIRS CORRESPOND TO THE
C     GIVEN COMMON DISTANCE.

1030 READ (10,510) TITLES
     READ (10,1040) (KSTH(I),I=1,180)
     READ (10,1050) (DIST25(I),I=1,180)
1040 FORMAT (3(60A4))
1050 FORMAT (3(60F10.4))
     DO 1060 I = 1, 180
       IF (NSTAT .EQ. KSTH(I)) NST = I
       IF (NSHOT .EQ. KSTH(I)) NSH = I
1060 CONTINUE
     X1 = (DIST25(NST) + DIST25(NSH)) / 2.0


C     STATION SPACING IS 2KM SO OMP SPACING IS 1KM <=> +/- 0.5KM !

1070 XU = X1 + 0.5
     XD = X1 - 0.5
     L = 0
     DO 1090 I = 1, 180
       NSTRT = I + 1
       IF (I .EQ. 180) NSTRT = NSTRT - 1
       DO 1090 J = NSTRT, 180
         XMP = (DIST25(I) + DIST25(J)) / 2.0
         IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 1090
         L = L + 1
         IST(L) = 0
         ISH(L) = 0
         DO 1080 IJ = 1, 110
           IF (KSTH(I) .EQ. ASTAT(IJ)) IST(L) = IJ
           IF (KSTH(J) .EQ. ASTAT(IJ)) IST(L) = IJ
           IF (IJ .GT. 71) GO TO 1080
           IF (KSTH(J) .EQ. ASHOT(IJ)) ISH(L) = IJ
           IF (KSTH(I) .EQ. ASHOT(IJ)) ISH(L) = IJ
1080     CONTINUE

C         IF IST(L) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C         RESET L AND IST AND ISH

         IF (IST(L) .NE. 0 .AND. ISH(L) .NE. 0) GO TO 1090
         IST(L) = 0
         ISH(L) = 0
         L = L - 1
1090 CONTINUE


C     TRACE POSITION GOT-TT ARE INPUTTED DIRECTLY     IQTT =0
C                        CORRECTION TO TT  ICORR=3 & IQTT =0
C                        CALCULATED FROM DISTANCES   IQTT =1
C                        CALCULATED FROM DIGIT FILE  IQTT =2
C
     NO = L
     L = 0
```

```
      DO 1150 I = 1, NO
        NSTAT = IST(I)
        NSHOT = ISH(I)
        IF (IQTT .EQ. 0) GO TO 1120
        IF (IQTT .EQ. 1) GO TO 1100

C        IQTT =2 : CALCULATE TT FROM UNIT 3

        CALL DIGIN(TOTT, T, ET)

C
C        IF BLANK LINE READ IN UNIT 3 RETURNS T<0.0
C        IF T<0.0 GO TO END OF TT CALCULATION AND INPUT SECTION
C        IF NOT CONTINUE
C        T=0.0 IF NO TT PICK MADE AT PRESENT SHOT
C
        IF (T .LE. 0.0) GO TO 1130
        T = T + TSTRT + XMAT(NSTAT,NSHOT) / VRED
        GO TO 1120

C        IQTT =1 : CALCULATE TT FROM SECTION MEASUREMENTS

1100    WRITE (6,1280) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
        READ (5,590) T, ET
        IF (T .LE. 0.0) GO TO 1130
        IF (T .EQ. 999.9 .AND. TT(NSTAT,NSHOT) .EQ. 0.0) GO TO 1150
        IF (T .EQ. 999.9) GO TO 1110
        T = (T/TSCALE) + TSTRT + XMAT(NSTAT,NSHOT) / VRED
        ET = ET / TSCALE
1110    GO TO 1130

C        IQTT =0 : INPUT TIME $ ERROR IN SECS

1120    WRITE (6,1270) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
        READ (5,590) T, ET
        IF (ICORR .EQ. 3) T = TT(NSTAT,NSHOT) + T / TSCALE
        IF (ICORR .EQ. 3) ET = ETT(NSTAT,NSHOT)

C        TT NOW CALCULATED :
C        INPUT THESE INTO ARRAYS BUT FIRST
C        COMPARE TO ORIGINAL VALUES IN MATRICES
C        IF DIFFERENT GIVE OPTIONS OF IGNORING
C        THE NEW OR OVERWRITING THE OLD VALUES
C
C        IF T=0.0 NO PICK MADE AT THIS SHOT          -GO TO NEXT TRACE
C        IF TT(NSTAT,I)=0.0  -ORIGINAL TT=0.0        -COPY NEW TT
C        IF TT(NSTAT,I)=T    -ORIGINAL TT= NEW TT    -GO TO NEXT TRACE
C        IF T=999.9          -SET TT=0.0             -GIVE OPTION TO
C                                                     COPY NEW TT
C        IF PASSES THROUGH ALL IF'S GIVE OPTION TO COPY NEW TT AS
C        T NOT EQUAL TO TT(NSTAT,I) AND SO NEW IS NOT EQUAL TO OLD TT.

1130    IF (T .LT. 0) GO TO 1290
        IF (TT(NSTAT,NSHOT) .EQ. 0.0) GO TO 1140
        IF (T .EQ. TT(NSTAT,NSHOT)) GO TO 1140
        IF (T .EQ. 999.9) T = 0.0
        IF (T .EQ. 999.9) ET = 0.0
        WRITE (6,950) ASTAT(NSTAT), ASHOT(NSHOT), TT(NSTAT,NSHOT)
        READ (5,270) IQ
        IF (IQ .EQ. 1) GO TO 1140
        WRITE (6,960)
        READ (5,270) IQ
        IF (IQ .EQ. 1) GO TO 1150

C        IF IQ=0 DO NOT GO TO NEXT STATION - JUMP OUT OF LOOP

        GO TO 1290

C
C                    ____________________
C                    COPY NEW TT TO ARRAY
C                    ____________________

1140    TT(NSTAT,NSHOT) = T
        ETT(NSTAT,NSHOT) = ET
        WRITE (7,640) ASTAT(NSTAT), ASHOT(NSHOT), T, ET
1150 CONTINUE
     GO TO 1290

C     GO TO END OF INPUT SECTION IF NO MORE TRACES REQUIRED

C
C*************** END TO COMMON MID-POINT SECTION ******************
C


C
C*****************************************************************
C*                                                               *
C*    COMMON MID-POINT : KIND=4                                  *
C*                                                               *
C*****************************************************************
C

1160 WRITE (6,1170)
1170 FORMAT (/'WHICH COMMON DISTANCE F10.3 ? '/'_____.___')
     READ (5,1180) CDS
1180 FORMAT (F10.3)

C     IQTT=2 READING DIGITISED FILE : IF SO THEN CHECK CORRECT
C                                     SHOT IS SPECIFIED


     IF (IQTT .NE. 2) GO TO 1190
     IF (CDSL .EQ. CDS) GO TO 1190
     WRITE (6,300)
     STOP
```

```
C     FIND TRACES' POSITIONS IN MATRICES

1190 XU = CDS + 1
     XD = CDS - 1
     L = 0
     DO 1200 I = 1, 110
       DO 1200 J = 1, 71
         IF (XMAT(I,J) .GT. XU .OR. XMAT(I,J) .LT. XD) GO TO 1200
         L = L + 1
         IST(L) = I
         ISH(L) = J
1200 CONTINUE


C     TRACE POSITIONS GOT-TT ARE INPUTTED DIRECTLY     IQTT =0
C                          CORRECTION TO TT  ICORR=3 & IQTT =0
C                          CALCULATED FROM DISTANCES   IQTT =1
C                          CALCULATED FROM DIGIT FILE  IQTT =2

     NO = L
     L = 0
     DO 1260 I = 1, NO
       NSTAT = IST(I)
       NSHOT = ISH(I)

       IF (IQTT .EQ. 0) GO TO 1230
       IF (IQTT .EQ. 1) GO TO 1210

C       IQTT =2 : CALCULATE TT FROM UNIT 3

       CALL DIGIN(DIGSCL, T, ET)

C       IF BLANK LINE READ IN UNIT 3 RETURNS T<0.0
C       IF T<0.0 GO TO END OF TT CALCULATION AND INPUT SECTION
C       IF NOT CONTINUE
C       T=0.0 IF NO TT PICK MADE AT PRESENT SHOT


       IF (T .LE. 0.0) GO TO 1240
       T = T + TSTRT + XMAT(NSTAT,NSHOT) / VRED
       GO TO 1240

C       IQTT =1 : READ IN DISTANCES AND USE TO CALCULATE TT

1210   WRITE (6,1280) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
       READ (5,590) T, ET
       IF (T .LE. 0.0) GO TO 1240
       IF (T .EQ. 999.9 .AND. TT(NSTAT,I) .EQ. 0.0) GO TO 1260
       IF (T .EQ. 999.9) GO TO 1220
       T = (T/TSCALE) + TSTRT + XMAT(NSTAT,NSHOT) / VRED
       ET = ET / TSCALE
1220   GO TO 1240


C       IQTT =0 : READ IN TT AND ERROR

1230   WRITE (6,1270) P, ASTAT(NSTAT), NSTAT, ASHOT(NSHOT), NSHOT
       READ (5,590) T, ET
       IF (ICORR .EQ. 3) T = TT(NSTAT,NSHOT) + T / TSCALE
       IF (ICORR .EQ. 3) ET = ETT(NSTAT,NSHOT)

C       TT NOW CALCULATED :
C       INPUT THESE INTO ARRAYS BUT FIRST
C       COMPARE TO ORIGINAL VALUES IN MATRICES
C       IF DIFFERENT GIVE OPTIONS OF IGNORING
C       THE NEW OR OVERWRITING THE OLD VALUES
C
C       IF T=0.0 NO PICK MADE AT THIS SHOT        -GO TO NEXT TRACE
C       IF TT(NSTAT,I)=0.0  -ORIGINAL TT=0.0      -COPY NEW TT
C       IF TT(NSTAT,I)=T    -ORIGINAL TT= NEW TT  -GO TO NEXT TRACE
C       IF T=999.9          -SET TT=0.0           -GIVE OPTION TO
C                                                  COPY NEW TT
C       IF PASSES THROUGH ALL IF'S GIVE OPTION TO COPY NEW TT AS
C       T NOT EQUAL TO TT(NSTAT,I) AND SO NEW IS NOT EQUAL TO OLD TT.


1240   IF (T .LT. 0) GO TO 1290
       IF (TT(NSTAT,NSHOT) .EQ. 0.0) GO TO 1250
       IF (T .EQ. TT(NSTAT,NSHOT)) GO TO 1250
       IF (T .EQ. 999.9) T = 0.0
       IF (T .EQ. 999.9) ET = 0.0
       WRITE (6,950) ASTAT(NSTAT), ASHOT(NSHOT), TT(NSTAT,NSHOT)
       READ (5,270) IQ
       IF (IQ .EQ. 1) GO TO 1250
       WRITE (6,960)
       READ (5,270) IQ
       IF (IQ .EQ. 1) GO TO 1260

C     GO TO END OF INPUT SECTION IF NO MORE TRACES REQUIRED

       GO TO 1290

C
C                         ____________________
C                         COPY NEW TT TO ARRAY
C                         ____________________

1250   TT(NSTAT,NSHOT) = T
       ETT(NSTAT,NSHOT) = ET
       WRITE (7,640) ASTAT(NSTAT), ASHOT(NSHOT), T, ET
1260 CONTINUE


C     GO TO END OF INPUT SECTION IF NO MORE TRACES REQUIRED

     GO TO 1290

C
C*************** END TO COMMON DISTANCE SECTION ******************
C
```

```
C------------------- FORMAT STATEMENTS --------------------

1270 FORMAT (1X, 'TT+ERR OR TTCORR ', A4, ' PHASE AT STATION ', A4,
    1        '(', I4, ') & SHOT ', A4, '(', I4, ')')
1280 FORMAT (1X, 'VAL(MM)+ ERR FOR ', A4, ' PHASE AT STATION ', A4,
    1        '(', I4, ') & SHOT ', A4, '(', I4, ')')
1290 WRITE (6,1300)
1300 FORMAT (/'CONTINUE WITH MORE INPUT OF THIS PHASE? YES=1 NO=0', /
    1        'TO UPDATE THIS MATRIX FILE WITH THE NEW DATA TYPE 3'/'_')
     READ (5,270) IO
     IF (IO .EQ. 1) GO TO 250

C     WRITE CHANGES TO DISC FILE

     REWIND 4
     WRITE (6,1310)
1310 FORMAT (/'UPDATING MATRIX FILE WITH NEW DATA')
     READ (4,510) TIN
     READ (4,1320) ADUMP, BDUMP
     READ (4,1320) ADUMP, BDUMP
1320 FORMAT (2A4)
     DO 1330 I = 1, 110
1330 WRITE (4,210) (TT(I,J),ETT(I,J),J=1,71)
     IF (IO .EQ. 3) GO TO 250
     WRITE (6,1340)
1340 FORMAT (/'DO YOU WANT TO INPUT NEW PHASE YES=1 NO=0')
     READ (5,270) IO
     IF (IO .NE. 1) GO TO 1350
     MEM = 1
     GO TO 20
C
1350 STOP
     END

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
     SUBROUTINE DIGIN(DIGSCL, T, ET)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     THIS SUBROUTINE USES THE CORNER POINTS OF THE SECTION
C     PLOTS AND THE DIGITISED PICK TO CALCULATE THE TT AND
C     ERROR.
C     FLAG = 2 : CORNER OF SECTION  : FIRST 4 LINES IN UNIT 3
C     FLAG = 1 : TT PICK
C     FLAG = D : ERROR IN TT PICK ie WHERE PICK MIGHT BE
C     FLAG = 7 : NO PICK MADE AT THIS TRACE
C                NB LEAVE NO GAPS IF A TRACE IS MISSINF STILL
C                INPUT FLAG = 7.
C     FLAG = 0 : EMPTY LINE AT END OF FILE WILL CAUSE A RETURN TO
C                MAIN PROGRAMME AND SET TT<0.0 TO SHOW THAT EOF
C                WAS REACHED.
C     MAIN PROGRAMME WILL GIVE NUMBER OF TRACES READ IF LESS THAN
C     THE NUMBER EXPECTED FOR THAT SECTION TYPE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
     INTEGER XP, YP, X1, Y1, X2, Y2
     COMMON X1, Y1, X2, Y2
     INTEGER*2 DD, D1, D7, ATYPE
     DATA DD /'D '/
     DATA D1 /'1 '/
     DATA D7 /'7 '/
     ICOUNT = 0
     T = 0.0
     ET = 0.0
  10 READ (3,20) ATYPE, XP, YP
  20 FORMAT (A2, I5, 1X, I5)
     WRITE (6,30) ATYPE, XP, YP
  30 FORMAT (1X, A2, I5, 1X, I5)
     IF (ATYPE .EQ. D7) RETURN
     IF (ATYPE .EQ. D1) GO TO 40
     IF (ATYPE .EQ. DD) GO TO 40
     T = -1.0
     RETURN
  40 XX = (YP - Y1)
     YY = (XP - X1)
     XY = XX / YY
     A = ATAN(XY)
     XX = (Y2 - Y1)
     YY = (X2 - X1)
     XY = XX / YY
     C = ATAN(XY)
     B = A - C
     D = ((XP - X1)**2 + (YP - Y1)**2) ** 0.5
     X = D * SIN(B)
     ICOUNT = ICOUNT + 1
     IF (ATYPE .EQ. D1) T = X * DIGSCL
     IF (ATYPE .EQ. DD) ET = X * DIGSCL
     IF (ICOUNT .LE. 1) GO TO 10
     ET = ABS(T - ET)
     RETURN
     END
C
```

```
C
CA    AHJ Lewis
C
CD+
C*********************************************************************
C                      PROGRAMME MATREAD
C*********************************************************************
C
C      READS TRAVEL TIME DATA IN THE TRAVEL TIME MATRICES WRITTEN BY
C      PROGRAMME MATCOPY.
C
C      UNIT 2 = DISTANCE MATRIX FOR M25B      )
C      UNIT 3 = DISTANCE MATRIX FOR CSSP      )    ASSIGNED TO FILES
C      UNIT 4 = TRAVEL TIME MATRIX            )    INTERNALLY
C      UNIT 5 )
C             ) = DEFAULT TO SCREEN
C      UNIT 6 )
C      UNIT 7 = OUTPUT OF TRAVEL TIME VALUES IN TABULATED FORMAT
C
C      $RUN MATREAD.C 7=OUTPUT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-
C
      INTEGER ASHOT, XSHOT, ASTAT, XSTAT
      DIMENSION TT(110,110), ETT(110,110), ASHOT(110),X(110,110),
     1          ASTAT(110), XSTAT(110), XSHOT(110), DIST25(180),
     2          KSTH(180), IST(110), ISH(110) , DCMP(110)
C
C
C*********************************************************************
C
C      ASSIGN DATA FILES TO UNIT NUMBERS AND READ DATA
C      -----------------------------------------------
C
      DATA NSEA /'N   '/
      DATA PG /'PG  '/
      DATA PC1 /'PC1 '/
      DATA PC2 /'PC2 '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PM1P /'PM1P'/
      DATA PM2P /'PM2P'/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
      DATA M25A /'M25A'/
      DATA M25B /'M25B'/
C
C      FIND OUT WHICH DATA IS TO BE INPUT
C
CCCCCCCCCCCC
C
C      MEM =0 FIRST TIME THROUGH PROGRAMME - READ ALL MATRICES
C      MEM =1 READ ONLY NEW ASSIGNED TT MATRIX TO UNIT 4
C
      MEM=0
C
CCCCCCCCCCCC
C
      WRITE(6,8)
    8 FORMAT(/'UNIT SPECIFICATION FOR MATCOPY IS :-'
     1//'ASSIGNED INTERNALLY:-'
     2/'UNIT 4  = TMAT-PHASE'
     3/'UNIT 5  = *SOURCE*'
     4/'UNIT 6  = *SOURCE'
     5/'UNIT 7  = LISTING OF TIMES'
     6/'UNIT 3  = GPT9:XMAT-CSSP'
     7/'UNIT 2  = GPT9:XMAT-M25B')
C
C
    9 WRITE (6,10)
   10 FORMAT (/'TRAVEL TIMES FOR WHICH PHASE? (EG. PG,PC1,PCP,PMP,PN)')
      READ (5,20) P
   20 FORMAT (A4)
      IF (P .EQ. PG) GO TO 30
      IF (P .EQ. PC1) GO TO 33
      IF (P .EQ. PC2) GO TO 32
      IF (P .EQ. PCP) GO TO 40
      IF (P .EQ. PMP) GO TO 50
      IF (P .EQ. PM1P) GO TO 51
      IF (P .EQ. PM2P) GO TO 52
      IF (P .EQ. PN) GO TO 60
      IF (P .EQ. PN1) GO TO 63
   30 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 70
   33 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC1', 22)
      GO TO 70
   32 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC2', 22)
      GO TO 70
   40 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 70
   50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 70
   51 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM1P', 23)
      GO TO 70
   52 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM2P', 23)
      GO TO 70
   60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 70
   63 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 70
   70 CONTINUE
      IF (MEM.EQ.1) GO TO 79
      CALL FTNCMD('ASSIGN 3=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 2=GPT9:XMAT-M25B', 23)
C
C      CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
```

```
C      IF NOT STOP
C
   79 READ (4,20) TIN
      IF (TIN .EQ. P) GO TO 90
      WRITE (6,80) TIN, P
   80 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
      STOP
C
C      READ IN VALUES ALREADY IN MATRIX FILE
C
   90 WRITE (6,100)
  100 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (4,110) (ASTAT(I),I=1,110)
      READ (4,110) (ASHOT(I),I=1,71)
      IF(MEM.EQ.1) GO TO 101
      READ (3,20) ADUMP
      READ (3,110) (XSTAT(I),I=1,110)
      READ (3,110) (XSHOT(I),I=1,71)
  110 FORMAT (110A4)
  101 DO 140 I = 1, 110
        READ (4,120) (TT(I,J),ETT(I,J),J=1,71)
      IF(MEM.EQ.1) GO TO 121
        READ (3,120) (X(I,J),J=1,71)
  120   FORMAT (71(F10.4,F10.4))
  121   IF (ASTAT(I) .EQ. XSTAT(I))GO TO 140
        IF(I.GT.71) GO TO 140
        IF(ASHOT(I) .EQ. XSHOT(I))GO TO 140
        WRITE (6,130)
  130   FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
        STOP
  140 CONTINUE
      READ (2,20) TITLES
      READ (2,591) (KSTH(I),I=1,180)
      READ (2,602) (DIST25(I),I=1,180)
  591 FORMAT (3(60A4))
  602 FORMAT (3(60F10.4))
      WRITE(6,601)
  601 FORMAT(/'DATA READ')
C
C      AFTER CHECKS
C
  200 WRITE (6,210)
  210 FORMAT (/'COMMON STATION =1 OR COMMON SHOT =2 OR CODS & ENDS= 3 OR
     1'/'COMMON MIDPOINT =4 OR COMMON DISTANCE =5 ? '/'_')
      READ (5,220) KIND
  220 FORMAT (I1)
  230 FORMAT (1X,'-----------------------------------------------------')
      IF (KIND .EQ. 1) GO TO 240
      IF (KIND .EQ. 2) GO TO 380
      IF (KIND .EQ. 3) GO TO 470
  231 IF (KIND .EQ. 4) GO TO 560
      IF (KIND .EQ. 5) GO TO 670
C
CCCCCCC
C
```

```
C      COMMON STATION
C
CCCCCC
C
  240 WRITE (6,250)
      WRITE (7,230)
  250 FORMAT (/'WHICH STATION - A4 ?'/'____')
      READ (5,260) ISTAT
  260 FORMAT (A4)
      WRITE (6,270)
  270 FORMAT (/'NORTH OR IRISH SEA? (N OR M)')
      READ (5,280) ISEA
  280 FORMAT (A4)
      IF (ISEA .EQ. NSEA) GO TO 290
      NSTRT = 1
      NEND = NSTRT + 36 - 1
      GO TO 300
  290 NSTRT = 37
      NEND = NSTRT + 30 - 1
  300 NST = 0
      DO 310 I = 1,110
  310 IF (ISTAT .EQ. ASTAT(I)) NST = I
      IF (NST .NE. 0) GO TO 330
      WRITE (6,320) ISTAT
  320 FORMAT (/'STAT ', A4, ' NOT FOUND')
      GO TO 730
  330 ISTAT = NST
      WRITE (7,340) P, ISEA, ASTAT(ISTAT)
  340 FORMAT (1X,' LISTING OF ', A4, ' PHASE IN ', A4,
     1        '-SEA FOR COMMON STATION ', A4,/
     2' STAT   SHOT  DIST      TT      ERROR     VRED=6  VRED=8   DISTM25B
     3       OMP DIST')
      DO 350 I = NSTRT, NEND
        T6 = 0.0
        T8 = 0.0
        IF (TT(ISTAT,I) .EQ. 0.0) GO TO 350
        T6 = TT(ISTAT,I) - X(ISTAT,I) / 6.0
        T8 = TT(ISTAT,I) - X(ISTAT,I) / 8.0
      DO 341 J = 1, 180
        IF (ASTAT(ISTAT).EQ. KSTH(J)) NST = J
        IF (ASHOT(I) .EQ. KSTH(J)) NSH = J
  341 CONTINUE
        XMP=(DIST25(NST)+DIST25(NSH))/2.0
        WRITE (7,361) ASTAT(ISTAT), ASHOT(I), X(ISTAT,I), TT(ISTAT,I),
     1  ETT(ISTAT,I), T6, T8 , DIST25(NST),DIST25(NSH),XMP
  350 CONTINUE
  360 FORMAT (2X, A4, 2X, A4, 7F10.3)
      WRITE (6,370)
  370 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER STAT? YES=1 NO=0',
     1        /'_')
      READ (5,220) IQ
      IF (IQ .EQ. 1) GO TO 240
      GO TO 730
C
CCCCCCC
```

```
C
C     COMMON SHOT
C
CCCCCCC
C
 380 WRITE (6,390)
     WRITE (7,230)
 390 FORMAT (/'WHICH SHOT (A4) ?'/'____')
     READ (5,400) ISHOT
 400 FORMAT (A4)
     NSH = 0
     DO 410 I = 1, 71
 410 IF (ISHOT .EQ. ASHOT(I)) NSH = I
     IF (NSH .NE. 0) GO TO 430
     WRITE (6,420) ISHOT
 420 FORMAT (/'SHOT ', A4, ' NOT FOUND')
     GO TO 730
 430 ISHOT = NSH
     WRITE (7,440) P,  ASHOT(ISHOT)
 440 FORMAT (1X,' LISTING OF ', A4, ' PHASE',
    1        ' FOR COMMON SHOT    ', A4, /
    3' STAT   SHOT  DIST     TT     ERROR    VRED=6  VRED=8   DISTM25B
    4        CMP DIST')
     DO 450 I = 1,110
       T6 = 0.0
       T8 = 0.0
       IF (TT(I,ISHOT) .EQ. 0.0) GO TO 450
       T6 = TT(I,ISHOT) - X(I,ISHOT) / 6.0
       T8 = TT(I,ISHOT) - X(I,ISHOT) / 8.0
     DO 441 J = 1, 180
       IF (ASTAT(I) .EQ. KSTH(J)) NST = J
       IF (ASHOT(ISHOT) .EQ. KSTH(J)) NSH = J
 441 CONTINUE
       XMP=(DIST25(NST)+DIST25(NSH))/2.0
       WRITE (7,361) ASTAT(I), ASHOT(ISHOT), X(I,ISHOT), TT(I,ISHOT),
    1  ETT(I,ISHOT), T6, T8,DIST25(NST),DIST25(NSH),XMP
 450 CONTINUE
     WRITE (6,460)
 460 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER SHOT? YES=1 NO=0',
    1        /'_')
     READ (5,220) IQ
     IF (IQ .EQ. 1) GO TO 380
     GO TO 730
C
CCCCCCC
C
C     CODS & BOBS
C
CCCCCCC
C
 470 WRITE (6,480)
     WRITE (7,230)
 480 FORMAT (/'WHICH STATION (A4)  & SHOT (A4)'/'____/____')
     READ (5,490) ISTAT, ISHOT
 490 FORMAT (A4, 1X, A4)
     NSH = 0
     NST = 0
     DO 500 I = 1,110
       IF (ISTAT .EQ. ASTAT(I)) NST = I
       IF (I .GT. 71) GO TO 500
 500 IF (ISHOT .EQ. ASHOT(I)) NSH = I
     IF (NSH .NE. 0) GO TO 520
     IF (NST .NE. 0) GO TO 520
     WRITE (6,510) ISHOT, ISTAT
 510 FORMAT (/'SHOT OR STATION ', A4, 1X, A4, ' NOT FOUND IN MATRIX')
     GO TO 730
 520 ISHOT = NSH
     ISTAT = NST
     WRITE (7,530) P, ASHOT(ISHOT), ASTAT(ISTAT)
 530 FORMAT (1X,' LISTING OF ', A4, ' PHASE FOR SHOT ',
    1        A4, ' STATION ', A4, /
    3' STAT   SHOT  DIST     TT     ERROR    VRED=6  VRED=8   DISTM25B
    4        CMP DIST')
     T6 = 0.0
     T8 = 0.0
     IF (TT(ISTAT,ISHOT) .EQ. 0.0) GO TO 540
     T6 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 6.0
     T8 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 8.0
     DO 531 J = 1, 180
       IF (ASTAT(ISTAT) .EQ. KSTH(J)) NST = J
       IF (ASHOT(ISHOT) .EQ. KSTH(J)) NSH = J
 531 CONTINUE
     XMP=(DIST25(NST)+DIST25(NSH))/2.0
     WRITE (7,361) ASTAT(ISTAT), ASHOT(ISHOT), X(ISTAT,ISHOT),
    1TT(ISTAT,ISHOT), ETT(ISTAT,ISHOT), T6, T8 ,DIST25(NST),
    2DIST25(NSH),XMP
 540 WRITE (6,550)
 550 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER VALUE? YES=1 NO=0'/
    1        '_')
     READ (5,220) IQ
     IF (IQ .EQ. 1) GO TO 470
     GO TO 730
C
CCCCCCCCCCCC
C
C     COMMON MID-POINT
C
CCCCCCCCCCCC
C
 560 WRITE (6,570)
 570 FORMAT (/'INPUT NEAREST STAT/SHOT PAIR TO SET CMP'/'____:____')
     WRITE(7,230)
     READ (5,580) ISTAT, ISHOT
 580 FORMAT (A4, 1X, A4)
     DO 610 I = 1, 180
       IF (ISTAT .EQ. KSTH(I)) NST = I
       IF (ISHOT .EQ. KSTH(I)) NSH = I
 610 CONTINUE
     X1 = (DIST25(NST) + DIST25(NSH)) / 2.0
 620 XU = X1 + 0.5
```

```
      XD = X1 - 0.5
      L = 0
      DO 640 I = 1, 180
        NSTRT = I + 1
        IF (I .EQ. 180) NSTRT = NSTRT - 1
        DO 640 J = NSTRT, 180
          XMP = (DIST25(I) + DIST25(J)) / 2.0
          IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 640
          L = L + 1
          IST(L) = 0
          ISH(L) = 0
          DO 630 IJ = 1,110
            IF (KSTH(I) .EQ. ASTAT(IJ)) IST(L) = IJ
            IF (KSTH(J) .EQ. ASTAT(IJ)) IST(L) = IJ
            IF (IJ .GT. 71) GO TO 630
            IF (KSTH(J) .EQ. ASHOT(IJ)) ISH(L) = IJ
            IF (KSTH(I) .EQ. ASHOT(IJ)) ISH(L) = IJ
 630      CONTINUE
          DOMP(L)=XMP
C
C          IF IST(L) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C          RESET L AND IST AND ISH
C
          IF (IST(L) .NE. 0 .AND. ISH(L) .NE. 0) GO TO 640
          IST(L) = 0
          ISH(L) = 0
          L = L - 1
 640  CONTINUE
      NO = L
      L = 0
      WRITE (7,650) P, ISTAT,ISHOT
 650  FORMAT (1X,' LISTING OF ', A4, ' PHASE FOR COMMON MID-POINT ', A4,
     1        '/', A4, /
     3' STAT   SHOT  DIST     TT     ERROR    VRED=6  VRED=8   DISTM25B
     4        CMP DIST')
      DO 660 I = 1, NO
        ISTAT = IST(I)
        ISHOT = ISH(I)
        T6 = 0.0
        T8 = 0.0
        IF (TT(ISTAT,ISHOT) .EQ. 0.0) GO TO 651
        T6 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 6.0
        T8 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 8.0
 651    WRITE (7,361) ASTAT(ISTAT), ASHOT(ISHOT), X(ISTAT,ISHOT),
     1  TT(ISTAT,ISHOT), ETT(ISTAT,ISHOT), T6, T8 ,DIST25(NST),
     2  DIST25(NSH),DOMP(I)
 361  FORMAT (2X, A4, 2X, A4, 8F8.3)
 660  CONTINUE
      GO TO 730
C
CCCCCCC
C
C       COMMON DISTANCE
C
CCCCCCC
 670  WRITE (6,680)
 680  FORMAT (/'INPUT COMMON DISTANCE F10.3'/'————.——')
      WRITE(7,230)
      READ (5,690) CDS
 690  FORMAT (F10.3)
      XU = CDS + 1.0
      XD = CDS - 1.0
      L = 0
      DO 700 I = 1,110
        DO 700 J = 1, 71
          IF (X(I,J) .GT. XU .OR. X(I,J) .LT. XD) GO TO 700
          IF (TT(I,J) .EQ. 0.0) GO TO 700
          L = L + 1
          IST(L) = I
          ISH(L) = J
 700  CONTINUE
      NO = L
      L = 0
      WRITE (7,710) P, CDS
 710  FORMAT (1X,' LISTING OF ', A4, ' PHASE FOR COMMON DISTANCE ',
     1        F10.4, /,
     3' STAT   SHOT  DIST     TT     ERROR    VRED=6  VRED=8   DISTM25B
     4        CMP DIST')
      DO 720 I = 1, NO
        ISTAT = IST(I)
        ISHOT = ISH(I)
        T6 = 0.0
        T8 = 0.0
        T6 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 6.0
        T8 = TT(ISTAT,ISHOT) - X(ISTAT,ISHOT) / 8.0
 711  DO 712 J = 1, 180
        IF (ASTAT(ISTAT).EQ. KSTH(J)) NST = J
        IF (ASHOT(ISHOT) EQ  KSTH(J)) NSH = J
 712  CONTINUE
      XCMP = (DIST25(NST) + DIST25(NSH)) / 2.0
        WRITE (7,361) ASTAT(ISTAT), ASHOT(ISHOT), X(ISTAT,ISHOT),
     1  TT(ISTAT,ISHOT), ETT(ISTAT,ISHOT), T6, T8 ,DIST25(NST),
     2  DIST25(NSH),XCMP
 720  CONTINUE
      GO TO 730
C
C      RETURN TO START IF MORE DATA AVAILABLE
C
C
 730        CONTINUE
 1010 WRITE(6,1020)
 1020 FORMAT (/'CONTINUE WITH MORE INPUT OF THIS PHASE? YES=1 NO=0')
      READ (5,220) IQ
      IF (IQ .EQ. 1) GO TO 200
      WRITE (6,1021)
 1021 FORMAT (/'DO YOU WANT TO INPUT NEW PHASE YES=1 NO=0')
      READ (5,220) IQ
      IF (IQ .NE. 1) GO TO 999
      GO TO 9
C
```

```
  999 STOP
      END
C
```

```
C
CA    AHJ Lewis
C
CD+
C*****************************************************************
C                    PROGRAMME TMATRD-CSH
C*****************************************************************
C
C     READS TRAVEL TIME DATA IN THE TRAVEL TIME MATRICES WRITTEN BY
C     PROGRAMME MATCOPY AND COPIES TRAVEL-TIMES for ALL COMMON SHOT
C     SECTIONS TO A DISC FILE WHICH
C     CAN BE USED AS INPUT FOR THE SEIS83 PROGRAMME GPT9:S83-RAYPLT
C
C     UNIT 2 = DISTANCE MATRIX FOR M25B      )
C     UNIT 3 = DISTANCE MATRIX FOR CSSP      )   ASSIGNED TO FILES
C     UNIT 4 = TRAVEL TIME MATRIX            )   INTERNALLY
C     UNIT 5 )
C            ) = DEFAULT TO SCREEN
C     UNIT 6 )
C     UNIT 7 = OUTPUT OF TRAVEL TIME VALUES IN TABULATED FORMAT
C
C     $RUN MATREAD.C 7=OUTPUT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-
C
      INTEGER ASHOT, XSHOT, ASTAT, XSTAT, AMAIN , AMAIN1(60)
      DIMENSION TT(110,110), ETT(110,110), ASHOT(110),X(110,110),
     1          ASTAT(110), XSTAT(110), XSHOT(110), DIST25(180),
     2          KSTH(180), IST(110), ISH(110) , DOMP(110),AMAIN(60),
     3          T(2000),XX(2000)
      DATA AMAIN /'N1  ', 'N2  ', 'N3  ', 'N4  ', 'N5  ', 'N6  ',
     1     'N7  ', 'N8  ', 'N9  ', 'N10 ', 'N11 ', 'N12 ', 'N13 ',
     2     'N14 ', 'N15 ', 'N16 ', 'N17 ', 'N18 ', 'N19 ', 'N20 ',
     3     'N21 ', 'N22 ', 'N23A', 'N23B', 'N24 ', 'N25 ', 'N26 ',
     4     'N27 ', 'N28 ', 'N29 ', 'M1  ', 'M2  ', 'M3  ', 'M4  ',
     5     'M5  ', 'M6  ', 'M7  ', 'M8  ', 'M9  ', 'M10 ', 'M11 ',
     6     'M12 ', 'M13 ', 'M14 ', 'M15 ', 'M16 ', 'M17 ', 'M18 ',
     7     'M19 ', 'M20 ', 'M21 ', 'M22 ', 'M23 ', 'M24 ', 'M25A',
     8     'M25B', 'I1A ', 'I1B '/
      DATA AMAIN1/'1   ', '2   ', '3   ', '4   ', '5   ', '6   ',
     1     '7   ', '8   ', '9   ', '10  ', '11  ', '12  ', '13  ',
     2     '14  ', '15  ', '16  ', '17  ', '18  ', '19  ', '20  ',
     3     '21  ', '22  ', '23  ', '24  ', '25  ', '26  ', '27  ',
     4     '28  ', '29  ', '30  ', '31  ', '32  ', '33  ', '34  ',
     5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
     6     '42  ', '43  ', '44  ', '45  ', '46  ', '47  ', '48  ',
     7     '49  ', '50  ', '51  ', '52  ', '53  ', '54  ', '55  ',
     8     '56  ', '57  ', '58  ','59  ','60  '/
C
C
C
C*****************************************************************
C
```

```
C     ASSIGN DATA FILES TO UNIT NUMBERS AND READ DATA
C     ------------------------------------------------
C
      DATA NSEA /'N   '/
      DATA PG /'PG  '/
      DATA PC1 /'PC1 '/
      DATA PC2 /'PC2 '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PM1P /'PM1P'/
      DATA PM2P /'PM2P'/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
      DATA M25A /'M25A'/
      DATA M25B /'M25B'/
C
C     FIND OUT WHICH DATA IS TO BE INPUT
C
      WRITE(6,8)
    8 FORMAT(/'UNIT SPECIFICATION FOR MATCOPY IS :-'
     1//'ASSIGNED INTERNALLY:-'
     2/'UNIT 4  = TMAT-PHASE'
     3/'UNIT 5  = *SOURCE*'
     4/'UNIT 6  = *SOURCE'
     5/'UNIT 7  = LISTING OF TIMES'
     6/'UNIT 3  = GPT9:XMAT-CSSP'
     7/'UNIT 2  = GPT9:XMAT-M25B')
C
C
CCCCCCCCCCCC
C
C     MEM =1 FIRST TIME THROUGH PROGRAMME - READ ALL MATRICES
C     MEM =2,3...    READ RELEVENT MATRICES TO LIST ALL DATA
C
      MEMX=0
      MEM=1
C
CCCCCCCCCCCC
C
C
  300 DO 905 NUM=1,70
      L=0
      WRITE(7,305) AMAIN(NUM)
      WRITE(6,305) AMAIN(NUM)
      WRITE(8,305) AMAIN(NUM)
  305 FORMAT(/'CSH/',A4)
      DO 900 MEM=1,5
      IF(MEM.EQ.1) P=PG
      IF(MEM.EQ.2) P=PCP
      IF(MEM.EQ.3) P=PMP
      IF(MEM.EQ.4) P=PN
      IF(MEM.EQ.5) P=PN1
      WRITE(6,306) P
  306 FORMAT(/'READING PHASE ',A4)
   20 FORMAT (A4)
      IF (P .EQ. PG) GO TO 30
      IF (P .EQ. PCP) GO TO 40
      IF (P .EQ. PMP) GO TO 50
      IF (P .EQ. PN) GO TO 60
      IF (P .EQ. PN1) GO TO 63
   30 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 70
   33 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC1', 22)
      GO TO 70
   32 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC2', 22)
      GO TO 70
   40 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 70
   50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 70
   51 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM1P', 23)
      GO TO 70
   52 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM2P', 23)
      GO TO 70
   60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 70
   63 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 70
   70 CONTINUE
      IF (MEMX.EQ.1) GO TO 79
      CALL FTNCMD('ASSIGN 3=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 2=GPT9:XMAT-M25B', 23)
C
C     CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
C     IF NOT STOP
C
   79 READ (4,20) TIN
      IF (TIN .EQ. P) GO TO 90
      WRITE (6,80) TIN, P
   80 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
      STOP
C
C     READ IN VALUES ALREADY IN MATRIX FILE
C
   90 WRITE (6,100)
  100 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (4,110) (ASTAT(I),I=1,110)
      READ (4,110) (ASHOT(I),I=1,71)
      IF(MEMX.EQ.1) GO TO 101
      READ (3,20) ADUMP
      READ (3,110) (XSTAT(I),I=1,110)
      READ (3,110) (XSHOT(I),I=1,71)
  110 FORMAT (110A4)
  101 DO 140 I=1,110
        READ (4,120) (TT(I,J),ETT(I,J),J=1,71)
      IF(MEMX.EQ.1) GO TO 121
        READ (3,120) (X(I,J),J=1,71)
  120   FORMAT (71(F10.4,F10.4))
```

```
 121    IF (ASTAT(I) .EQ. XSTAT(I))GO TO 140
        IF(I.GT.71) GO TO 140
        IF(ASHOT(I) .EQ. XSHOT(I))GO TO 140
        WRITE (6,130)
 130    FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
        STOP
 140 CONTINUE
     IF(MEMX.EQ.1) GO TO 603
     READ (2,20) TITLES
     READ (2,591) (KSTH(I),I=1,180)
     READ (2,602) (DIST25(I),I=1,180)
 591 FORMAT (3(60A4))
 602 FORMAT (3(60F10.4))
 603 WRITE(6,601)
 601 FORMAT(/'DATA READ')
C
C     AFTER CHECKS
C
C
     ISHOT=AMAIN(NUM)
     DO 310 I = 1,66
 310 IF (ISHOT .EQ. ASHOT(I)) NSH = I
     IF (NSH .NE. 0) GO TO 330
     WRITE (6,320) ISHOT
 320 FORMAT (/'SHOT ', A4, ' NOT FOUND')
     GO TO 900
 330 ISHOT = NSH
     DO 350 I = 1,110
     NST=0
       DO 331 J = 1,60
 331     IF (ASTAT(I) .EQ. AMAIN1(J)) NST = J
       IF (NST .EQ. 0) GO TO 350
       T6 = 0.0
       T8 = 0.0
       IF (TT(I,ISHOT) .EQ. 0.0) GO TO 350
       T6 = TT(I,ISHOT) - X(I,ISHOT) / 6.0
       T8 = TT(I,ISHOT) - X(I,ISHOT) / 8.0
     DO 341 J = 1, 180
       IF (ASTAT(I).EQ. KSTH(J)) NST = J
       IF (ASHOT(ISHOT) .EQ. KSTH(J)) NSH = J
 341 CONTINUE
       XMP=(DIST25(NST)+DIST25(NSH))/2.0
       WRITE (7,360) ASTAT(I), ASHOT(ISHOT), X(I,ISHOT), TT(I,ISHOT),
    1  ETT(I,ISHOT), T6, T8 , DIST25(NST),DIST25(NSH),XMP,P
     L=L+1
     T(L)=TT(I,ISHOT)
     XX(L)=DIST25(NST)
 350 CONTINUE
 360 FORMAT (2X, A4, 2X, A4, 8F10.3,2X,A4)
C
900        CONTINUE
           WRITE(8,362) L
           WRITE(8,361) ((XX(I),T(I)),I=1,L)
361        FORMAT((8F10.5))
362        FORMAT(I3)
905        CONTINUE
C
 999 STOP
     END
C
```

```
C
CA    AHJ Lewis
C
CD+
C*********************************************************************
C                        PROGRAMME TMATRD-CST
C*********************************************************************
C
C     READS TRAVEL TIME DATA IN THE TRAVEL TIME MATRICES WRITTEN BY
C     PROGRAMME MATCOPY FOR ALL COMMON STATIONS ONTO DISC IN A
C     FORMAT WHICH CAN BE READ BY THE SEIS83 PROGRAMME GPT9:S83-RAYPLT.
C
C     UNIT 2 = DISTANCE MATRIX FOR M25B       )
C     UNIT 3 = DISTANCE MATRIX FOR CSSP       )   ASSIGNED TO FILES
C     UNIT 4 = TRAVEL TIME MATRIX             )   INTERNALLY
C     UNIT 5 )
C            ) = DEFAULT TO SCREEN
C     UNIT 6 )
C     UNIT 7 = OUTPUT OF TRAVEL TIME VALUES IN TABULATED FORMAT
C
C     $RUN MATREAD.C 7=OUTPUT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CD-
C
      INTEGER ASHOT, XSHOT, ASTAT, XSTAT, AMAIN
      DIMENSION TT(110,110), ETT(110,110), ASHOT(110),X(110,110),
     1          ASTAT(110), XSTAT(110), XSHOT(110), DIST25(180),T(2000),
     2          KSTH(180), IST(110), ISH(110) , DCMP(110),AMAIN(110),
     3          XX(2000)
C     DATA AMAIN /'2   ', '4   ', '5   ', '6   ', '7   ', '8   ',
C    1   2 '11  ', '16  ', '19  ', '23  ', '25  ', '26  ', '27  ',
C    2           '30  ', '31  ',  '32  ', '33  ', '34  ',
C    5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
C    6     '42  ', '43  ', '44  ', '45  ', '46  ', '47  ', '48  ',
C    7     '49  ', '50  ', '51  ', '53  ', '54  ','52  ','55  ',
C    8     '56  ','57  ','58  ',  '59  ', '60  ','31N ','31S '/
      DATA AMAIN /'1   ', '2   ', '3   ', '4   ', '5   ', '6   ',
     1     '7   ', '8   ', '11  ', '15  ', '19  ', '23  ', '25  ',
     2     '26  ','27  ', '30  ', '31  ',  '32  ', '33  ', '34  ',
     5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
     6     '42  ',  '44  ', '45  ', '46  ', '47  ', '48  ',
     7     '49  ', '50  ', '51  ','52  ', '53  ', '54  ','55  ',
     8     '56  ','57  ','58  ','59  ', '60  ','31N ','31S '/
C
C
C*********************************************************************
C
C     ASSIGN DATA FILES TO UNIT NUMBERS AND READ DATA
C     ________________________________________________
C
      DATA NSEA /'N   '/
      DATA PG /'PG  '/
      DATA PC1 /'PC1 '/
      DATA PC2 /'PC2 '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PM1P /'PM1P'/
      DATA PM2P /'PM2P'/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
      DATA M25A /'M25A'/
      DATA M25B /'M25B'/
C
C      FIND OUT WHICH DATA IS TO BE INPUT
C
      WRITE(6,8)
    8 FORMAT(/'UNIT SPECIFICATION FOR MATCOPY IS :-'
     1//'ASSIGNED INTERNALLY:-'
     2/'UNIT 4  = TMAT-PHASE'
     3/'UNIT 5  = *SOURCE*'
     4/'UNIT 6  = *SOURCE'
     5/'UNIT 7  = LISTING OF TIMES'
     6/'UNIT 3  = GPT9:XMAT-CSSP'
     7/'UNIT 2  = GPT9:XMAT-M25B')
C
C
CCCCCCCCCCC
C
C      MEM =1 FIRST TIME THROUGH PROGRAMME - READ ALL MATRICES
C      MEM =2,3.       READ RELEVENT MATRICES TO LIST ALL DATA
C
      MEMX=0
      MEM=1
C
CCCCCCCCCCC
C
      NSTRT = 1
      NEND =  66
C
  300 DO 905 NUM=1,70
      L=0
      WRITE(7,305) AMAIN(NUM)
      WRITE(6,305) AMAIN(NUM)
      WRITE(8,305) AMAIN(NUM)
  305 FORMAT(/'CST/',A4)
      DO 900 MEM=1,5
      IF(MEM.EQ.1) P=PG
      IF(MEM.EQ.2) P=PCP
      IF(MEM.EQ.3) P=PMP
      IF(MEM.EQ.4) P=PN
      IF(MEM.EQ.5) P=PN1
      WRITE(6,306) P
  306 FORMAT(/'READING PHASE ',A4)
   20 FORMAT (A4)
      IF (P .EQ. PG) GO TO 30
```

```
      IF (P .EQ. PCP) GO TO 40
      IF (P .EQ. PMP) GO TO 50
      IF (P .EQ. PN) GO TO 60
      IF (P .EQ. PN1) GO TO 63
   30 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 70
   33 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC1', 22)
      GO TO 70
   32 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PC2', 22)
      GO TO 70
   40 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 70
   50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 70
   51 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM1P', 23)
      GO TO 70
   52 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM2P', 23)
      GO TO 70
   60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 70
   63 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 70
   70 CONTINUE
      IF (MEMX.EQ.1) GO TO 79
      CALL FTNCMD('ASSIGN 3=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 2=GPT9:XMAT-M25B', 23)
C
C      CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
C      IF NOT STOP
C
   79 READ (4,20) TIN
      IF (TIN .EQ. P) GO TO 90
      WRITE (6,80) TIN, P
   80 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
      STOP
C
C      READ IN VALUES ALREADY IN MATRIX FILE
C
   90 WRITE (6,100)
  100 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (4,110) (ASTAT(I),I=1,110)
      READ (4,110) (ASHOT(I),I=1,71)
      IF(MEMX.EQ.1) GO TO 101
      READ (3,20) ADUMP
      READ (3,110) (XSTAT(I),I=1,110)
      READ (3,110) (XSHOT(I),I=1,71)
  110 FORMAT (110A4)
  101 DO 140 I=1,110
        READ (4,120) (TT(I,J),ETT(I,J),J=1,71)
      IF(MEMX.EQ.1) GO TO 121
        READ (3,120) (X(I,J),J=1,71)
  120   FORMAT (71(F10.4,F10.4))
  121   IF (ASTAT(I) .EQ. XSTAT(I))GO TO 140
        IF(I.GT.71) GO TO 140
        IF(ASHOT(I) .EQ. XSHOT(I))GO TO 140
        WRITE (6,130)
  130   FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
        STOP
  140 CONTINUE
      IF(MEMX.EQ.1) GO TO 603
      READ (2,20) TITLES
      READ (2,591) (KSTH(I),I=1,180)
      READ (2,602) (DIST25(I),I=1,180)
  591 FORMAT (3(60A4))
  602 FORMAT (3(60F10.4))
  603 WRITE(6,601)
  601 FORMAT(/'DATA READ')
C
C      AFTER CHECKS
C
C
      ISTAT=AMAIN(NUM)
      DO 310 I = 1,110
  310 IF (ISTAT .EQ. ASTAT(I)) NST = I
      IF (NST .NE. 0) GO TO 330
      WRITE (6,320) ISTAT
  320 FORMAT (/'STAT ', A4, ' NOT FOUND')
      GO TO 900
  330 ISTAT = NST
      DO 350 I = NSTRT, NEND
        T6 = 0.0
        T8 = 0.0
        IF (TT(ISTAT,I) .EQ. 0.0) GO TO 350
        T6 = TT(ISTAT,I) - X(ISTAT,I) / 6.0
        T8 = TT(ISTAT,I) - X(ISTAT,I) / 8.0
      DO 341 J = 1, 180
        IF (ASTAT(ISTAT).EQ. KSTH(J)) NST = J
        IF (ASHOT(I) .EQ. KSTH(J)) NSH = J
  341 CONTINUE
        XMP=(DIST25(NST)+DIST25(NSH))/2.0
        WRITE (7,360) ASTAT(ISTAT), ASHOT(I), X(ISTAT,I), TT(ISTAT,I),
     1  ETT(ISTAT,I), T6, T8 , DIST25(NST),DIST25(NSH),XMP,P
        L=L+1
        T(L) = TT(ISTAT,I)
        XX(L) = DIST25(NSH)
  350 CONTINUE
  360 FORMAT (2X, A4, 2X, A4, 8F10.3,2X,A4)
C
900       CONTINUE
        WRITE(8,362) L
        WRITE (8,361) ((XX(I), T(I)),I=1,L)
  362 FORMAT(I3)
  361 FORMAT ((8F10.5))
905       CONTINUE
C
  999 STOP
      END
C
```

```
C
CA    AHJ Lewis
C
CD+
C                    DISTCOP
C
C
C
C     THIS PROGRAMME COPIES INPUTTED DISTANCES INTO THE DISTANCE MATRIX
C     ASSIGNED TO UNIT 8.
C     DISTREAD WILL SIMILARLEY READ THE INPUTTED VALUES BACK AGAIN AS A
C     CHECK.
C
C     UNITS 5 & 6 = DEFAULT TO SCREEN
C     UNIT 8 = DIST MATRIX GPT9:XMAT-CSSP
C
C
CD-
C
C
      INTEGER ASHOT, ASTAT, XSTAT, XSHOT, X1, X2, X3, X4, Y1, Y2, Y3,
     1        Y4, KIND
      DIMENSION ASHOT(110), ASTAT(110), XMAT(110,110), KSTH(180),
     1          DIST25(180), IST(110), ISH(110), XSTAT(110), XSHOT(110),
     2 DOMP(110)
      DATA NSEA /'N   '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PG /'PG  '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
C
C     FIND OUT WHICH DATA IS TO BE INPUT
C
C
C     READ IN VALUES ALREADY IN MATRIX FILE
C
   10 WRITE (6,20)
   20 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (8,170) ADUMP
      READ (8,30) (ASTAT(I),I=1,110)
      READ (8,30) (ASHOT(I),I=1,71)
   30 FORMAT (110A4)
      DO 50 I = 1, 110
        READ (8,40) (XMAT(I,J),J=1,71)
   40   FORMAT (71(F10.4,F10.4))
   50 CONTINUE
C
C
   60 WRITE (6,70)
   70 FORMAT (/'COMMON STATION =1 OR COMMON SHOT =2')
      READ (5,80) KIND
   80 FORMAT (I1)
      IF (KIND .EQ. 1) GO TO 90
      IF (KIND .EQ. 2) GO TO 260
C
CCCCCCC
C
C     COMMON STATION
C
CCCCCC
C
   90 WRITE (6,100)
  100 FORMAT (/'WHICH STATION A4 ?'/'____')
      READ (5,110) NSTAT
  110 FORMAT (A4)
  120 NST = 0
      DO 130 I = 1, 110
  130 IF (NSTAT .EQ. ASTAT(I)) NST = I
      IF (NST .NE. 0) GO TO 150
      WRITE (6,140) NSTAT
  140 FORMAT (/'STATION  NOT FOUND IN MATRIX ', A4)
      GO TO 380
  150 NSTAT = NST
      WRITE (6,160)
  160 FORMAT (/'NORTH OR IRISH SEA? (N OR M)'/'_')
      READ (5,170) ISEA
  170 FORMAT (A4)
      IF (ISEA .EQ. NSEA) GO TO 180
      NSTRT = 1
      NEND = NSTRT + 36 - 1
      GO TO 190
  180 NSTRT = 37
      NEND = NSTRT + 30 - 1
  190 DO 250 I = NSTRT, NEND
  200   WRITE (6,210) ASTAT(NSTAT), ASHOT(I)
  210   FORMAT (1X, 'DIST FOR STAT ', A4, '  SHOT ', A4, ' ?')
        READ (5,220) X
  220   FORMAT (F10.4)
  230   XMAT(NSTAT,I) = X
        WRITE (7,240) ASTAT(NSTAT), ASHOT(I), X
  240   FORMAT (/'DIST FOR ', A4, ' AND ', A4, ' INPUT IS ', F10.3)
  250 CONTINUE
      GO TO 380
C
CCCCCCC
C
C     COMMON SHOT
C
CCCCCCC
C
  260 WRITE (6,270)
  270 FORMAT (/'WHICH SHOT (A4) ?'/'____')
      READ (5,280) NSHOT
  280 FORMAT (A4)
  290 NSH = 0
      DO 300 I = 1, 71
  300 IF (NSHOT .EQ. ASHOT(I)) NSH = I
      IF (NSH .NE. 0) GO TO 320
```

```
      WRITE (6,310) NSHOT
  310 FORMAT (/'SHOT ', A4, ' NOT FOUND')
      GO TO 380
  320 NSHOT = NSH
      DO 360 I = 1, 110
  330   WRITE (6,210) ASTAT(I), ASHOT(NSHOT)
        READ (5,340) X
  340   FORMAT (F10.4)
  350   XMAT(I,NSHOT) = X
        WRITE (7,240) ASTAT(I), ASHOT(NSHOT), X
  360 CONTINUE
      WRITE (6,370)
  370 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER SHOT? YES=1 NO=0',
     1        /'_')
      READ (5,80) IO
      IF (IO .EQ. 1) GO TO 260
      GO TO 380
C
  380 WRITE (6,390)
  390 FORMAT ('TO CONTINUE =1'/, 'TO UPDATE THIS MATRIX FILE WITH THE NE
     1W DATA TYPE 3'/'_')
      READ (5,80) IO
      IF (IO .EQ. 1) GO TO 60
C
C      WRITE CHANGES TO DISC FILE
C
      REWIND 8
      WRITE (6,400)
  400 FORMAT (/'UPDATING MATRIX FILE WITH NEW DATA')
      READ (8,170) TIN
      READ (8,410) ADUMP, BDUMP
      READ (8,410) ADUMP, BDUMP
  410 FORMAT (2A4)
      DO 420 I = 1, 110
  420 WRITE (8,40) (XMAT(I,J),J=1,71)
      IF (IO .EQ. 3) GO TO 60
      STOP
      END
C
```

```
C
CA    AHJ Lewis
C
C                    DISTREAD
C                    --------
C
C     THIS PROGRAMME READS DISTANCES FROM THE DISTANCE MATRIX
C     ASSIGNED TO UNIT 8.
C     DISTCOPY WILL SIMILARLEY PUT THE VALUES INTO THE MATRICES
C
C     UNITS 5 & 6 = DEFAULT TO SCREEN
C     UNIT 8 = DIST MATRIX GPT9:XMAT-CSSP
C     UNIT 7 = OUTPUT
C
CD-
C
      INTEGER ASHOT, ASTAT, XSTAT, XSHOT, X1, X2, X3, X4, Y1, Y2, Y3,
     1        Y4, KIND
      DIMENSION ASHOT(110), ASTAT(110), XMAT(110,110), KSTH(180),
     1 DOMP(110),DIST25(180), IST(110), ISH(110), XSTAT(110), XSHOT(110)
      DATA NSEA /'N   '/
      DATA POP /'POP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PG /'PG  '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
C
C     FIND OUT WHICH DATA IS TO BE INPUT
C
C
C     READ IN VALUES ALREADY IN MATRIX FILE
C
   10 WRITE (6,20)
   20 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (8,170) ADUMP
      READ (8,30) (ASTAT(I),I=1,110)
      READ (8,30) (ASHOT(I),I=1,71)
   30 FORMAT (110A4)
      DO 50 I = 1, 110
        READ (8,40) (XMAT(I,J),J=1,71)
   40   FORMAT (71(F10.4,F10.4))
   50 CONTINUE
      READ(3,170) ADUMP
      READ(3,151) (KSTH(I),I=1,180)
      READ(3,152) (DIST25(I),I=1,180)
  151 FORMAT(180A4)
  152 FORMAT(180F10.4)
C
C
   60 WRITE (6,70)
   70 FORMAT (/'COMMON STATION =1 OR COMMON SHOT =2 , OMP =3, CDS =4')
      READ (5,80) KIND
   80 FORMAT (I1)
      IF (KIND .EQ. 1) GO TO 90
```

```
      IF (KIND .EQ. 2) GO TO 260
      IF (KIND .EQ. 3) GO TO 560
      IF (KIND .EQ. 4) GO TO 670
C
CCCCCCC
C
C     COMMON STATION
C
CCCCCC
C
   90 WRITE (6,100)
  100 FORMAT (/'WHICH STATION A4 ?'/'____')
      READ (5,110) NSTAT
  110 FORMAT (A4)
  120 NST = 0
      DO 130 I = 1, 110
  130 IF (NSTAT .EQ. ASTAT(I)) NST = I
      IF (NST .NE. 0) GO TO 150
      WRITE (6,140) NSTAT
  140 FORMAT (/'STATION  NOT FOUND IN MATRIX ', A4)
      GO TO 380
  150 NSTAT = NST
      WRITE (6,160)
  160 FORMAT (/'NORTH OR IRISH SEA? (N OR M)'/'_')
      READ (5,170) ISEA
  170 FORMAT (A4)
      IF (ISEA .EQ. NSEA) GO TO 180
      NSTRT = 1
      NEND = NSTRT + 36 - 1
      GO TO 190
  180 NSTRT = 37
      NEND = NSTRT + 30 - 1
  190 DO 250 I = NSTRT, NEND
  200   WRITE (7,210) ASTAT(NSTAT), ASHOT(I),XMAT(NSTAT,I)
  210   FORMAT (1X,'DIST FROM STAT ', A4,' TO SHOT ', A4,' IS ',F10.4)
  250 CONTINUE
      GO TO 380
C
CCCCCCC
C
C     COMMON SHOT
C
CCCCCCC
C
  260 WRITE (6,270)
  270 FORMAT (/'WHICH SHOT (A4) ?'/'____')
      READ (5,280) NSHOT
  280 FORMAT (A4)
  290 NSH = 0
      DO 300 I = 1, 71
  300 IF (NSHOT .EQ. ASHOT(I)) NSH = I
      IF (NSH .NE. 0) GO TO 320
      WRITE (6,310) NSHOT
  310 FORMAT (/'SHOT ', A4, ' NOT FOUND')
      GO TO 380
  320 NSHOT = NSH
      DO 360 I = 1, 110
  330   WRITE (7,210) ASTAT(I), ASHOT(NSHOT),XMAT(I,NSHOT)
  360 CONTINUE
      WRITE (6,370)
  370 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER SHOT? YES=1 NO=0',
     1        /'_')
      READ (5,80) IQ
      IF (IQ .EQ. 1) GO TO 260
      GO TO 380
C
CCCCCCCCCCCC
C
C     COMMON MID-POINT
C
CCCCCCCCCCCC
C
  560 WRITE (6,570)
  570 FORMAT (/'INPUT NEAREST STAT/SHOT PAIR TO SET OMP'/'____:____')
      WRITE(6,20)
      READ (5,580) ISTAT, ISHOT
  580 FORMAT (A4, 1X, A4)
      DO 610 I = 1, 180
        IF (ISTAT .EQ. KSTH(I)) NST = I
        IF (ISHOT .EQ. KSTH(I)) NSH = I
  610 CONTINUE
      X1 = (DIST25(NST) + DIST25(NSH)) / 2.0
  620 XU = X1 + 0.5
      XD = X1 - 0.5
      L = 0
      DO 640 I = 1, 180
        NSTRT = I + 1
        IF (I .EQ. 180) NSTRT = NSTRT - 1
        DO 640 J = NSTRT, 180
          XMP = (DIST25(I) + DIST25(J)) / 2.0
          IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 640
          L = L + 1
          IST(L) = 0
          ISH(L) = 0
          DO 630 IJ = 1,110
            IF (KSTH(I) .EQ. ASTAT(IJ)) IST(L) = IJ
            IF (KSTH(J) .EQ. ASTAT(IJ)) IST(L) = IJ
            IF (IJ .GT. 71) GO TO 630
            IF (KSTH(J) .EQ. ASHOT(IJ)) ISH(L) = IJ
            IF (KSTH(I) .EQ. ASHOT(IJ)) ISH(L) = IJ
  630     CONTINUE
          DOMP(L)=XMP
C
C          IF IST(I) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C          RESET L AND IST AND ISH
C
          IF (IST(L) .NE. 0 .AND. ISH(L) .NE. 0) GO TO 640
          IST(L) = 0
          ISH(L) = 0
          L = L - 1
```

```
 640 CONTINUE
     NO = L
     L = 0
     WRITE (7,650)  ISTAT,ISHOT
 650 FORMAT (1X,' LISTING OF DIST FOR COMMON MID-POINT ', A4,' / ',A4,
    1/'  STAT        SHOT        DIST       DIST M25B        DIST OMP')
     DO 660 I = 1, NO
       ISTAT = IST(I)
       ISHOT = ISH(I)
 651   WRITE (7,361) ASTAT(ISTAT), ASHOT(ISHOT), XMAT(ISTAT,ISHOT),
    1                DIST25(ISTAT), DIST25(ISHOT),DOMP(I)
 361 FORMAT (2X, A4, 2X, A4, 4F8.3)
 660 CONTINUE
     GO TO 380
C
CCCCCCC
C
C      COMMON DISTANCE
C
CCCCCCC
 670 WRITE (6,680)
 680 FORMAT (/'INPUT COMMON DISTANCE F10.3'/'————.——')
     WRITE(6,20)
     READ (5,690) CDS
 690 FORMAT (F10.3)
     XU = CDS + 1
     XD = CDS - 1
     L = 0
     DO 700 I = 1,110
       DO 700 J = 1, 71
         IF (XMAT(I,J) .GT. XU .OR. XMAT(I,J) .LT. XD) GO TO 700
         L = L + 1
         IST(L) = I
         ISH(L) = J
 700 CONTINUE
     NO = L
     L = 0
     WRITE (7,710)  CDS
 710 FORMAT (1X,' LISTING OF DIST FOR COMMON DISTANCE ',F10.4, /,
    3' STAT   SHOT  DIST       DISTM25B     OMP DIST')
     DO 720 I = 1, NO
       ISTAT = IST(I)
       ISHOT = ISH(I)
 711 DO 712 J = 1, 180
       IF (ASTAT(ISTAT).EQ. KSTH(J)) NST = J
       IF (ASHOT(ISHOT).EQ. KSTH(J)) NSH = J
 712 CONTINUE
     XOMP = (DIST25(NST) + DIST25(NSH)) / 2.0
       WRITE (7,361) ASTAT(ISTAT), ASHOT(ISHOT), XMAT(ISTAT,ISHOT),
    1  DIST25(ISTAT),DIST25(ISHOT),XOMP
 720 CONTINUE
     GO TO 380
C
 380 WRITE (6,390)
 390 FORMAT ('TO CONTINUE =1')
     READ (5,80) IQ
     IF (IQ .EQ. 1) GO TO 60
     STOP
     END
C
```

```
C
CA      AHJ Lewis
C
CD+
C                    DISTCOPM25
C
C     copies distances of all CSSP shots and stations from M25B
C
C     UNITS 5 & 6 = DEFAULT TO SCREEN
C     UNIT 8      = DIST MATRIX GPT9:XMAT-M25B
C
CD-
C
      INTEGER ASHOT, ASTAT, KIND
      DIMENSION ASTAT(180), XMAT(180)
      DATA NSEA /'N   '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PG /'PG  '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
C
C     FIND OUT WHICH DATA IS TO BE INPUT
C     READ IN VALUES ALREADY IN MATRIX FILE
C
   10 WRITE (6,20)
   20 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (8,30) ADUMP
   30 FORMAT (A4)
      READ (8,40) (ASTAT(I),I=1,180)
      READ (8,50) (XMAT(I),I=1,180)
   40 FORMAT (3(60A4))
   50 FORMAT (3(60F10.4))
C
   60 WRITE (6,70)
   70 FORMAT (/'WHICH STATION OR SHOT  A4 ?'/'____')
      READ (5,80) NSTAT
   80 FORMAT (A4)
   90 NST = 0
      DO 100 I = 1, 180
  100 IF (NSTAT .EQ. ASTAT(I)) NST = I
      IF (NST .NE. 0) GO TO 120
      WRITE (6,110) NSTAT
  110 FORMAT (/'STATION/SHOT NOT FOUND IN MATRIX ', A4)
      GO TO 170
  120 NSTAT = NST
  130 WRITE (6,140) ASTAT(NSTAT)
  140 FORMAT (1X, 'DIST FOR STAT ', A4, '?')
      READ (5,150) X
  150 FORMAT (F10.4)
      XMAT(NSTAT) = X
      WRITE (7,160) ASTAT(NSTAT), X
  160 FORMAT (/'DIST FOR ', A4, ' INPUT IS ', F10.3)
      GO TO 170
C
  170 WRITE (6,180)
  180 FORMAT ('TO CONTINUE =1'/, 'TO UPDATE THIS MATRIX FILE WITH THE NE
     1W DATA TYPE 3'/'_')
      READ (5,190) IQ
  190 FORMAT (I1)
      IF (IQ .EQ. 1) GO TO 60
C
C      WRITE CHANGES TO DISC FILE
C
      REWIND 8
      WRITE (6,200)
  200 FORMAT (/'UPDATING MATRIX FILE WITH NEW DATA')
      READ (8,30) TIN
  210 WRITE (8,40) (ASTAT(J),J=1,180)
      WRITE (8,50) (XMAT(J),J=1,180)
      STOP
      END
C
```

```
C
CA     AHJ LEWIS
C
CD+
C                    DISTRD.M25B
C
C     READS DISTANCES HELD IN GPT9:XMAT-M25B
C
C     UNITS 5 & 6 = DEFAULT TO SCREEN
C     UNIT 8      = DIST MATRIX GPT9:XMAT-M25B
C
C
CD-
C
      INTEGER ASHOT, ASTAT, KIND
      DIMENSION ASTAT(180), XMAT(180)
      DATA NSEA /'N   '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PG /'PG  '/
      DATA N23A /'N23A'/
      DATA N23B /'N23B'/
C
C      FIND OUT WHICH DATA IS TO BE INPUT
C      READ IN VALUES ALREADY IN MATRIX FILE
C
   10 WRITE (6,20)
   20 FORMAT (/'READING DATA FROM MATRIX FILE')
      READ (8,30) ADUMP
   30 FORMAT (A4)
      READ (8,40) (ASTAT(I),I=1,180)
      READ (8,50) (XMAT(I),I=1,180)
   40 FORMAT (3(60A4))
   50 FORMAT (3(60F10.4))
C
      DO 60 NSTAT = 1, 180
   60 WRITE (7,70) ASTAT(NSTAT), XMAT(NSTAT)
   70 FORMAT (1X, 'DIST FROM  ', A4, ' TO M25B IS  ', F10.3, ' KM')
      STOP
      END
C
```

```
C
CA    AHJ LEWIS
CD+
C ***********************************************************************
C                    PROGRAMME TT-REG
C***********************************************************************
C
C  PROGRAM TO PERFORM A LINEAR LEAST SQUARES FIT TO TRAVEL TIMES TO
C  LAYERS GIVING WIDE ANGLE REFLECTIONS AND/OR A REFRACTOR PHASES.
C  REFLECTIONS INTERPRETATED USING A T**2-X**2 PLOT
C  REFRACTORS USING A NORMAL T-X PLOT
C
C  PLOTS IN A4 SIZE CAN BE OBTAINED BY LEAVING SCALING CARD BLANK
C
C ---------------------------------------------------------------------
C -------
C  $R *FTNX SCARDS=GLL4:TT-REG  SPUNCH=-T
C  $R -T+GPT9:TAUP+*GHOST+*NAG 5=INPUT  7=OUTPUT  9=PLOTFILE
C ---------------------------------------------------------------------
C -------
C
C    INPUT
C    -----
C    1) NORTH OR IRISH SEA(N,M),PLOT TITLE   (SIN,TITLE  A4,19A4)
C    2) MAINLINE=1 ALL=0, STATION NO, SHOT NO, CDS DISTANCE
C       (I5,2A4,F10.3)
C    3) NO. OF FIRST AND LAST POINTS FOR LEAST SQ (NST,NEND) FOR
C       REFRACTOR  AND REFLECTOR TIMES
C       IF TAUP VALUES REQUIRED PUT 1 (I5) HERE OTHERWISE LEAVE BLANK
C       4* (2I5) (3LINES)
C    4) MIN,MAX SLOWNESS AND NUMBER OF RAY PARAMETERS TO BE
C       CONSIDERED(3F10.0)
C    5) MIN,MAX DISTANCE AND DIST INCREMENT FOR INTERPOLATION
C    6) REDUCTION VELOCITY (0.0=UNREDUCED) OF INPUT DATA & OF PLOT
C       (2F10.0)
C    7) TIME SCALE(CM/SEC) DIST SCALE(CM/KM) ,P SCALE TAU SCALE
C       (TSCALE,DSCALE,PSCALE,TAUSCA,4F10.0)
C    8) XMIN,XMAX,TMIN,TMAX  4(F5.2)
C
C
C       UNITS
C       -----
C
C       UNIT 1-4 ,8 & 10 ARE ASSIGNED INTERNALLY TO THE TRAVEL TIME
C       MATRICES
C        AS FOLLOWS:-
C        1 = PN, 2 = PMP, 3 = PCP, 4 = PG, 8&10 = DISTANCE MATRIX
C
C       UNIT 7 IS THE MAIN OUTPUT
C       UNIT 6 DEFAULTS TO THE SCREEN
C       UNIT 9 CONTAINS THE PLOTTER OUTPUT
C
C    ORIGINAL PROGRAMME BY A.H.J.LEWIS + A. GREEN
C    TAU-P ROUTINES VIA CAMBRIDGE -WRITTEN BY B.L.N. KENNET
C
```

```
CD-
      REAL*8 RSULTN(20), RSULTM(20), RSULTC(20), TT(200), XX(200),
     1       RSULTG(20)
      INTEGER ASTAT, ASHOT, AMAIN, ITIME(5), BLANK
      DIMENSION TN(200), TC(200), TM(200), X(200), D(200), TITLES(20),
     1          ETN(200), ETC(200), ETM(200), TG(200), ETG(200),
     2          TIM(200), XDIS(200), Y(2), XA(2), XP(200), ET(200),
     3          TP(200), TITLE(20), XXA(200), YY(200), TCOR(200),
     4          T(200), XT(200), PBKN(200), TAUKN(200), PBBN(200),
     5          TPMAXN(200), TPMINN(200), PBKM(200), TAUKM(200),
     6          PBBM(200), TPMAXM(200), TPMINM(200), PBKC(200),
     7          TAUKC(200), PBBC(200), TPMAXC(200), TPMINC(200),
     8          PBKG(200), TAUKG(200), PBBG(200), TPMAXG(200),
     9          TPMING(200), ASTAT(110), ASHOT(110), IST(110), ISH(110),
     *          KSTH(180), DIST25(180), AMAIN(60)
      COMMON /TT/ XMAT(110,110), TTN(110,110), TTM(110,110),
     1       TTC(110,110), TTG(110,110)
      COMMON /ETT/ ETTN(110,110), ETTM(110,110), ETTC(110,110),
     1       ETTG(110,110)
      COMMON /P/ VAV, EVAV, VN, EVN, ZM, EZM, ZN, EZN, XCRIT, EXCRIT,
     1       RSULTM, RSULTN
      COMMON /PC/ VC, EVC, ZC, EZC, RSULTC
      COMMON /G/ VG, EVG, ZG, EZG, RSULTG
      COMMON /TP/ TP, XP, PO, PM, FP, XO, XM, DX, TITLE
      COMMON /RED/ ASTAT, ASHOT, ISTAT, ISHOT, IST, ISH
      DATA NSEA /'N   '/
      DATA BLANK/'    '/
      DATA AMAIN /'1   ', '2   ', '3   ', '4   ', '5   ', '6   ',
     1     '7   ', '8   ', '9   ', '10  ', '11  ', '12  ', '13  ',
     2     '14  ', '15  ', '16  ', '17  ', '18  ', '19  ', '20  ',
     3     '21  ', '22  ', '23  ', '24  ', '25  ', '26  ', '27  ',
     4     '28  ', '29  ', '30  ', '31  ', '32  ', '33  ', '34  ',
     5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
     6     '42  ', '43  ', '44  ', '45  ', '46  ', '47  ', '48  ',
     7     '49  ', '50  ', '51  ', '52  ', '53  ', '54  ', '55  ',
     8     '56  ', '57  ', '58  ', '59  ', '60  '/
C
CCCCCCCCCCCCCCCCCCCC
C
C  INPUT  OF DATA
C
CCCCCCCCCCCCCCCCCCCC
C
      VN=0.0
      ZN=0.0
      VC=0.0
      ZC=0.0
      VAV=0.0
      ZM=0.0
      WRITE(12,1)
 1    FORMAT(1X,' DIST M25B        PMP              PN              PCP')
      MEMRED = 0
 10   MEMN = 0
      MEMM = 0
      MEMC = 0
      MEMG = 0
      MEMTPN = 0
      MEMTPM = 0
      MEMTPC = 0
      MEMTPG = 0
      DO 20 I = 1, 110
        TN(I) = 0.0
        TM(I) = 0.0
        TC(I) = 0.0
 20   TG(I) = 0.0
      WRITE (6,30)
 30   FORMAT (/'NORTH(N) OR IRISH(M) SEA (A4) ? & TITLE FOR OUTPUT'/
     1        'N/M?-___TITLE_____ETC ')
      READ (5,40) ISEA, TITLE
 40   FORMAT (21A4)
      WRITE (7,40) TITLE
      WRITE (11,40) TITLE
      WRITE (6,50)
 50   FORMAT (/'ALL STATIONS =0 - MAINLINE ONLY =1,'/
     1        'STAT, SHOT (2A4   )COMMON DISTANCE (F10.3)  '//
     2        'COMMON STATION - LEAVE SHOT BLANK'/
     3        'CCOMMON SHOT - LEAVE STATION BLANK'/
     4        'COMMON MID-POINT - TYPE IN STAT/SHOT'/
     5        'COMMON DIST. - TYPE IN DISTANCE ONLY'//
     6        '_/////////____/____/_______.___')
      READ (5,60) MAINL, ISTAT, ISHOT, CDS
 60   FORMAT (I1, 8X, A4, 1X, A4, 1X, F10.3)
 70   FORMAT (2I5, I1)
 80   FORMAT (3I5)
 90   FORMAT (3F10.0)
      WRITE (6,100)
100   FORMAT (/'INPUT RANGE IN TT ARRAY OVER WHICH TO PERFORM REGRESSION
     1'/'FOR PN,PMP,PCP&PG IN TURN'/
     2        'NOTE LEAVE BOTH VALUES=0 IF P-PHASE  IS NOT REQUIRED'/
     3
     4'IF TAU-P REQUIRED TYPE 1 FOR LAST VALUE IF NOT  LEAVE BLANK'/
     5        '_____-_____- FOR PN ?')
      READ (5,70) NSTN, NENDN, MEMTPN
      WRITE (6,110)
110   FORMAT (/'_____-_____- FOR PMP ?')
      READ (5,70) NSTM, NENDM, MEMTPM
      WRITE (6,120)
120   FORMAT (/'_____-_____- FOR PCP ?')
      READ (5,70) NSTC, NENDC, MEMTPC
      WRITE (6,130)
130   FORMAT (/'_____-_____- FOR PG ?')
      READ (5,70) NSTG, NENDG, MEMTPG
      WRITE (6,140)
140   FORMAT (/'INPUT MIN & MAX SLOWNESS,AND NUMBER OF RAY PARAMETRES TO
     1 CONSIDER FOR TAU-P'/'_____.___-_____.___-_____.___')
      READ (5,90) PO, PM, FP
      WRITE (6,150)
150   FORMAT (/'INPUT MIN & MAX DISTANCE,AND DIST INCREM. FOR INTERPOLAT
     1 ION FOR TAU-P'/'_____.___-_____.___-_____.___')
      READ (5,90) XO, XM, DX
```

```
C
      IF (NSTG .EQ. NENDG) MEMG = 1
      IF (NSTC .EQ. NENDC) MEMC = 1
      IF (NSTM .EQ. NENDM) MEMM = 1
      IF (NSTN .EQ. NENDN) MEMN = 1
      WRITE (6,160)
  160 FORMAT (/'INPUT RED. VEL. OF INPUT AND OF FINAL PLOT'/
     1        '_______.___________.___')
      READ (5,170) VREDD, VREDP
  170 FORMAT (2F10.0)
  180 FORMAT (4F10.0)
      WRITE (6,190)
  190 FORMAT (/'TASCALE,DSCALE,PSCALE,TASCALE'/'_______.___________.___
     1___.___________.___')
      READ (5,180) TSCALE, DSCALE, PSCALE, TASCAL
      WRITE (6,200)
  200 FORMAT (/'XMIN,XMAX,TMIN,TMAX FOR T-X PLOT'/'LEAVE BLANK FOR AUTOM
     1ATIC SCALING'/'_______.___________.___________.___________.___')
      READ (5,180) XMIN, XMAX, TMIN, TMAX
C
      WRITE (6,40) TITLE
      WRITE (6,60) MAINL, ISTAT, ISHOT, CDS
      WRITE (6,80) NSTN, NENDN, MEMTPN
      WRITE (6,80) NSTM, NENDM, MEMTPM
      WRITE (6,80) NSTC, NENDC, MEMTPC
      WRITE (6,80) NSTG, NENDG, MEMTPG
      WRITE (6,90) PO, PM, FP
      WRITE (6,90) XO, XM, DX
      WRITE (6,180) TSCALE, DSCALE, PSCALE, TASCAL
      WRITE (6,210) MEMG, MEMC, MEMM, MEMN
  210 FORMAT (/'MEMG,MEMC,MEMM,MEMN', 4I5)
      WRITE (6,170) VREDD, VREDP
      WRITE (6,180) XMIN, XMAX, TMIN, TMAX
C
      IF (MEMRED .EQ. 1) GO TO 215
      CALL FTNCMD('ASSIGN 1=GPT9:TMAT-PN', 21)
      CALL FTNCMD('ASSIGN 2=GPT9:TMAT-PMP', 22)
      CALL FTNCMD('ASSIGN 3=GPT9:TMAT-PCP', 22)
      CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      CALL FTNCMD('ASSIGN 8=GPT9:XMAT-CSSP', 23)
      CALL FTNCMD('ASSIGN 10=GPT9:XMAT-M258', 24)
C
C
C
      READ (8,40) TITLES
      READ (8,260) (ASTAT(I),I=1,110)
      READ (8,260) (ASHOT(I),I=1,71)
  215 NSH = 0
      NST = 0
      DO 230 I = 1, 110
        IF (ISTAT .EQ. BLANK .AND. ISHOT .EQ. BLANK) GO TO 240
        IF (ISTAT .EQ. BLANK) GO TO 220
        IF (ISTAT .EQ. ASTAT(I)) NST = I
  220   IF (ISHOT .EQ. BLANK) GO TO 230
        IF (I  GT. 71) GO TO 230
        IF (ISHOT .EQ. ASHOT(I)) NSH = I
  230 CONTINUE
      ISTAT = NST
      ISHOT = NSH
      IF (ISTAT .NE. 0) NST = ASTAT(ISTAT)
      IF (ISHOT .NE. 0) NSH = ASHOT(ISHOT)
      IF (ISTAT .EQ. 0) NST = ISEA
      IF (ISHOT .EQ. 0) NSH = ISEA
      WRITE(6,239) ISTAT,ISHOT
  239 FORMAT(2I5)
  240 WRITE (6,250) NST, NSH, CDS
  250 FORMAT (/'STAT/SHOT ARE', 2X, A4, 1X, A4, ' COMMON DIST IS ',
     1        F10.4)
  260 FORMAT (110A4)
  270 FORMAT (2A4)
  280 FORMAT (I1)
      IF (MEMRED .EQ. 1) GO TO 480
  290 WRITE (6,300)
  300 FORMAT (/'DATA BEING READ IN FROM MATRIX FILES')
C
      READ (1,310) T1, S1
      READ (2,310) T2, S2
      READ (3,310) T3, S3
      READ (4,310) T4, S4
C
      READ (1,310) T1, S1
      READ (2,310) T2, S2
      READ (3,310) T3, S3
      READ (4,310) T4, S4
C
      READ (1,310) T1, S1
      READ (2,310) T2, S2
      READ (3,310) T3, S3
      READ (4,310) T4, S4
C
  310 FORMAT (2A4)
  320 FORMAT (I1)
      M = 0
  330 DO 430 I = 1, 110
        NSHOT = 71
        IF (MAINL .EQ. 0) GO TO 390
        IF(M.LT.60) GO TO 331
        M=M+1
        ASTAT(M)=NSEA
        GO TO 430
  331   L = 0
        DO 340 K = 1, 60
          IF (ASTAT(I) .EQ. AMAIN(K)) L = K
  340   CONTINUE
        IF (L .EQ. 0) GO TO 350
        NSHOT = 66
        GO TO 390
  350   READ (8,310) T1, S1S
        IF (MEMG .NE. 0) GO TO 360
        READ (4,310) T1, S1
```

```
 360   IF (MEMC .NE. 0) GO TO 370
       READ (3,310) T1, S1
 370   IF (MEMM .NE. 0) GO TO 380
       READ (2,310) T1, S1
 380   IF (MEMN .NE. 0) GO TO 430
       READ (1,310) T1, S1
       GO TO 430
 390   M = M + 1
       IF(ASTAT(I).EQ.BLANK) GO TO 430
       ASTAT(M) = ASTAT(I)
       IF (ASTAT(ISTAT) .EQ. ASTAT(M)) ISTAT = M
       READ (8,450) (XMAT(M,J),J=1,NSHOT)
       IF (MEMG .NE. 0) GO TO 400
       READ (4,440) (TTG(M,J),ETTG(M,J),J=1,NSHOT)
 400   IF (MEMC .NE. 0) GO TO 410
       READ (3,440) (TTC(M,J),ETTC(M,J),J=1,NSHOT)
 410   IF (MEMM .NE. 0) GO TO 420
       READ (2,440) (TTM(M,J),ETTM(M,J),J=1,NSHOT)
 420   IF (MEMN .NE. 0) GO TO 430
       READ (1,440) (TTN(M,J),ETTN(M,J),J=1,NSHOT)
 430 CONTINUE
 440 FORMAT (71(2F10.4))
 450 FORMAT (71F10.4)
     READ (10,40) TITLES
     READ (10,460) (KSTH(I),I=1,180)
     READ (10,470) (DIST25(I),I=1,180)
 460 FORMAT (180A4)
 470 FORMAT (180F10.4)
 480 IF (ISTAT .NE. 0 .AND. ISHOT .EQ. 0) GO TO 500
     IF (ISTAT .EQ. 0 .AND. ISHOT .NE. 0) GO TO 580
     IF (ISTAT .NE. 0 .AND. ISHOT .NE. 0) GO TO 640
     IF (CDS .NE. 0) GO TO 750
     WRITE (6,490)
 490 FORMAT (/'INPUT FILE INCORRECT')
     STOP
C
CCCCCCCCC
C
C        COMMON STATION
C
CCCCCCCCC
C
 500 WRITE (7,510) ASTAT(ISTAT)
     WRITE (11,510) ASTAT(ISTAT)
 510 FORMAT (////'COMMON STATION SECTION ', A4, '-----------------------------
    1-----------------------------'/)
     IF (ISEA .EQ. NSEA) GO TO 520
     NSTRT = 1
     NEND = NSTRT + 36 - 1
     GO TO 530
 520 NSTRT = 37
     NEND = NSTRT + 30 - 1
 530 L = 0
     DO 570 I = NSTRT, NEND
       L = L + 1
       IST(L) = ISTAT
       ISH(L) = I
       X(L) = XMAT(ISTAT,I)
       IF (MEMN .NE. 0) GO TO 540
       TN(L) = TTN(ISTAT,I)
       ETN(L) = ETTN(ISTAT,I)
 540   IF (MEMM .NE. 0) GO TO 550
       TM(L) = TTM(ISTAT,I)
       ETM(L) = ETTM(ISTAT,I)
 550   IF (MEMC .NE. 0) GO TO 560
       TC(L) = TTC(ISTAT,I)
       ETC(L) = ETTC(ISTAT,I)
 560   IF (MEMG .NE. 0) GO TO 570
       TG(L) = TTG(ISTAT,I)
       ETG(L) = ETTG(ISTAT,I)
 570 CONTINUE
     NMOD = L
     GO TO 840
C
CCCCCCCCC
C
C        COMMON SHOT
C
CCCCCCCCC
C
 580 WRITE (7,590) ASHOT(ISHOT)
     WRITE (11,590) ASHOT(ISHOT)
 590 FORMAT (////'COMMON SHOT SECTION ', A4, '-----------------------------
    1-----------------------------'/)
     NEND = 110
     IF (MAINL .EQ. 1) NEND = 60
     L = 0
     DO 630 I = 1, NEND
       L = L + 1
       IST(L) = I
       ISH(L) = ISHOT
       X(I) = XMAT(I,ISHOT)
C
       IF (MEMN .NE. 0) GO TO 600
       TN(I) = TTN(I,ISHOT)
       ETN(I) = ETTN(I,ISHOT)
 600   IF (MEMM .NE. 0) GO TO 610
       TM(I) = TTM(I,ISHOT)
       ETM(I) = ETTM(I,ISHOT)
 610   IF (MEMC .NE. 0) GO TO 620
       TC(I) = TTC(I,ISHOT)
       ETC(I) = ETTC(I,ISHOT)
 620   IF (MEMG .NE. 0) GO TO 630
       TG(I) = TTG(I,ISHOT)
       ETG(I) = ETTG(I,ISHOT)
 630 CONTINUE
     NMOD = 110
     IF (MAINL .EQ. 1) NMOD = 60
     GO TO 840
C
```

```
CCCCCCCCCCCCC
C
C       CREATE SUBRPOINT
C
CCCCCCCCCCCCC
C
 640 WRITE (7,650) ASTAT(ISTAT), ASHOT(ISHOT)
     WRITE (11,650) ASTAT(ISTAT), ASHOT(ISHOT)
 650 FORMAT (////'COMMON MIDPOINT SECTION ', A4, '/', A4, /
    1
    2'————————————————————————————————————————————————————————'/)
     DO 660 I = 1, 160
       IF (ASTAT(ISTAT) .EQ. KSTH(I)) NST = I
       IF (ASHOT(ISHOT) .EQ. KSTH(I)) NSH = I
 660 CONTINUE
     X1 = (DIST25(NST) + DIST25(NSH)) / 2.0
 670 XU = X1 + 0.5
     XD = X1 - 0.5
     L = 0
     DO 740 I = 1, 160
       NSTRT = I + 1
       IF (I .EQ. 160) NSTRT = NSTRT - 1
       DO 740 J = NSTRT, 160
         XMP = (DIST25(I) + DIST25(J)) / 2.0
         IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 740
         L = L + 1
         IST(L) = 0
         ISH(L) = 0
         NEND = 110
         IF (MAINL .EQ. 1) NEND = 60
         DO 690 IJ = 1, NEND
           IF (KSTH(I) .EQ. ASTAT(IJ)) IST(L) = IJ
           IF (KSTH(J) .EQ. ASTAT(IJ)) IST(L) = IJ
 680       IF (IJ .GT. 71) GO TO 690
           IF (KSTH(J) .EQ. ASHOT(IJ)) ISH(L) = IJ
           IF (KSTH(I) .EQ. ASHOT(IJ)) ISH(L) = IJ
 690     CONTINUE
C
C         IF IST(L) OR ISH(L) IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C         RESET L AND IST AND ISH
C
         IF (IST(L) .EQ. 0 .OR. ISH(L) .EQ. 0) GO TO 730
         M = IST(L)
         K = ISH(L)
         X(L) = XMAT(M,K)
         IF (MEMN .NE. 0) GO TO 700
         TN(L) = TTN(M,K)
         ETN(L) = ETTN(M,K)
 700     IF (MEMM .NE. 0) GO TO 710
         TM(L) = TTM(M,K)
         ETM(L) = ETTM(M,K)
 710     IF (MEMC .NE. 0) GO TO 720
         TC(L) = TTC(M,K)
         ETC(L) = ETTC(M,K)
 720     IF (MEMG .NE. 0) GO TO 740
         TG(L) = TTG(M,K)
         ETG(L) = ETTG(M,K)
         GO TO 740
 730     IST(L) = 0
         ISH(L) = 0
         L = L - 1
 740 CONTINUE
     NMCG = L
     GO TO 840
C
CCCCCCC
C
C       COMMON DISTANCE
C
CCCCCCC
 750 WRITE (7,760) CDS
     WRITE (11,760) CDS
 760 FORMAT (////'COMMON DISTANCE SECTION ', F10.3, /'————————————
    1————————————————————'/)
     XU = CDS + 1
     XD = CDS - 1
     L = 0
     NEND = 110
     NEND1 = 71
     IF (MAINL .EQ. 1) NEND = 60
     IF (MAINL .EQ. 1) NEND1 = 66
     DO 770 I = 1, NEND
       DO 770 J = 1, NEND1
         IF (XMAT(I,J) .GT. XU .OR. XMAT(I,J) .LT. XD) GO TO 770
         L = L + 1
         IST(L) = I
         ISH(L) = J
 770 CONTINUE
     NO = L
     L = 0
     DO 810 I = 1, NO
       M = IST(I)
       J = ISH(I)
       X(I) = XMAT(M,J)
       IF (MEMN .NE. 0) GO TO 780
       TN(I) = TTN(M,J)
       ETN(I) = ETTN(M,J)
 780   IF (MEMM .NE. 0) GO TO 790
       TM(I) = TTM(M,J)
       ETM(I) = ETTM(ISTAT,I)
 790   IF (MEMC .NE. 0) GO TO 800
       TC(I) = TTC(M,J)
       ETC(I) = ETTC(M,J)
 800   IF (MEMG .NE. 0) GO TO 810
       TG(I) = TTG(M,J)
       ETG(I) = ETTG(M,J)
 810 CONTINUE
     NMCG = NO
C
CCCCCCCCCCCCCC
```

```
C
C     IF COMMON DISTANCE PLOT REQUIRED NEED TO CHANGE DISTANCES
C     TO DIST FROM M25B OF EACH PARTICULAR STAT/SHOT'S  MID-POINT
C
CCCCCCCCCCCCCCC
C
      DO 830 I = 1, NMCG
        M = IST(I)
        K = ISH(I)
        DO 820 J = 1, 160
          IF (ASTAT(M) .EQ. KSTH(J)) NST = J
          IF (ASHOT(K) .EQ. KSTH(J)) NSH = J
 820    CONTINUE
        X(I) = (DIST25(NST) + DIST25(NSH)) / 2.0
 830 CONTINUE
C
CCCCCCCCCCCCCCCCCC
C
C     OUTPUT OF TT AND DISTS
C
CCCCCCCCCCCCCCCCCCC
C
      GO TO 840
 840  IF (MEMN .EQ. 1) GO TO 850
      ISIG = 2
      CALL REDUC(NMCG, VREDP, X, TN, ISIG)
 850  IF (MEMM .EQ. 1) GO TO 860
      ISIG = 3
      CALL REDUC(NMCG, VREDP, X, TM, ISIG)
 860  IF (MEMC .EQ. 1) GO TO 870
      ISIG = 4
      CALL REDUC(NMCG, VREDP, X, TC, ISIG)
 870  IF (MEMG .EQ. 1) GO TO 880
      ISIG = 5
      CALL REDUC(NMCG, VREDP, X, TG, ISIG)
 880  CONTINUE
      IF (NENDG .GT. NMCG) NENDG = NMCG
      IF (NENDC .GT. NMCG) NENDC = NMCG
      IF (NENDM .GT. NMCG) NENDM = NMCG
      IF (NENDN .GT. NMCG) NENDN = NMCG
      IF (CDS .NE. 0.0) GO TO 1120
      GO TO 890
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     FIT T**2-X**2 TO REFLECTION DATA
C
C     ***IF MEMTPM=1 CALCULATE TAUP***
C     ***IF MEMTPC=1 CALCULATE TAUP***
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PMP REFLECTIONS
C
CCCCCC
C
 890  IF (MEMM .EQ. 1) GO TO 940
      DO 900 I = 1, NMCG
        TT(I) = 0.0
        T(I) = 0.0
        ET(I) = 0.0
        XT(I) = 0.0
 900  XX(I) = 0.0
      L = 0
      DO 910 I = NSTM, NENDM
        IF (TM(I) .EQ. 0.0) GO TO 910
        L = L + 1
        IF (L .EQ. 1) MSTM = I
        MENDM=I
        TT(L) = TM(I) ** 2
        XX(L) = X(I) ** 2
        T(L) = TM(I)
        ET(L) = ETM(I)
        XT(L) = X(I)
 910  CONTINUE
      IF (L .LE. 2) MEMM = 1
      IF (MEMM .EQ. 1) GO TO 940
      WRITE (11,920)
 920  FORMAT (//'************ T*T-X*X  PMP-REGRESSION ************'/)
      CALL REG(L, XX, TT, RSULTM)
      IF (MEMTPM .EQ. 0) GO TO 940
      NTAUKM = 0
      NTAUBM = 0
      WRITE (11,930)
 930  FORMAT (/'---------------  TAU-P PROCESSING ---------------'/)
      CALL TAUP(L, NTAUKM, NTAUBM, T, XT, ET, PBKM, TAUKM, PBBM, TPMAXM,
     1      TPMINM)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PCP REFLECTIONS
C
CCCC
C
 940  IF (MEMC .EQ. 1) GO TO 980
      DO 950 I = 1, NMCG
        TT(I) = 0.0
        T(I) = 0.0
        ET(I) = 0.0
        XT(I) = 0.0
 950  XX(I) = 0.0
      L = 0
      DO 960 I = NSTC, NENDC
        IF (TC(I) .EQ. 0.0) GO TO 960
        L = L + 1
        IF (L .EQ. 1) MSTC = I
        MENDC=I
        TT(L) = TC(I) ** 2
        T(L) = TC(I)
```

```
      ET(L) = ETC(I)
      XX(L) = X(I) ** 2
      XT(L) = X(I)
  960 CONTINUE
      IF (L .LE. 2) MEMC = 1
      IF (MEMC .EQ. 1) GO TO 980
      WRITE (11,170)
  970 FORMAT (//'************** T*T-X*X  POP-REGRESSION *************'/)
      CALL REG(L, XX, TT, RSULTC)
      IF (MEMTPC .EQ. 0) GO TO 980
      NTAUKC = 0
      NTAUBC = 0
      WRITE (11,930)
      CALL TAUP(L, NTAUKC, NTAUBC, T, XT, ET, PBKC, TAUKC, PBBC, TPMAXC,
     1      TPMINC)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     FIT T-X TO REFRACTION DATA
C
C     **IF MEMTPN=1 COMP. TAUP**
C     **IF MEMTPG=1 COMP. TAUP**
C
CCCCC
C
  980 IF (MEMN .EQ. 1) GO TO 1020
      DO 990 I = 1, NMOG
        TT(I) = 0.0
        T(I) = 0.0
        ET(I) = 0.0
        XT(I) = 0.0
  990 XX(I) = 0.0
      L = 0
      DO 1000 I = NSTN, NENDN
        IF (TN(I) .EQ. 0.0) GO TO 1000
        L = L + 1
        IF (L .EQ. 1) MSTN = I
        MENDN=I
        TT(L) = TN(I)
        T(L) = TN(I)
        ET(L) = ETN(I)
        XT(L) = X(I)
        XX(L) = X(I)
 1000 CONTINUE
      IF (L .LE. 2) MEMN = 1
      IF (MEMN .EQ. 1) GO TO 1020
      WRITE (11,1010)
 1010 FORMAT (//'*************** T-X REGRESSION ***************'/)
      CALL REG(L, XX, TT, RSULTN)
      IF (MEMTPN .EQ. 0) GO TO 1020
      NTAUKN = 0
      NTAUBN = 0
      WRITE (11,930)
      CALL TAUP(L, NTAUKN, NTAUBN, T, XT, ET, PBKN, TAUKN, PBBN, TPMAXN,
     1      TPMINN)
C
CCCCCCCCCCCCCCCC
C
C     PG REGRESSION
C
CCCCC
C
 1020 IF (MEMG .EQ. 1) GO TO 1050
      DO 1030 I = 1, NMOG
        TT(I) = 0.0
        T(I) = 0.0
        ET(I) = 0.0
        XT(I) = 0.0
 1030 XX(I) = 0.0
      L = 0
      DO 1040 I = NSTG, NENDG
        IF (TG(I) .EQ. 0.0) GO TO 1040
        L = L + 1
        IF (L .EQ. 1) MSTG = I
        MENDG=I
        TT(L) = TG(I)
        T(L) = TG(I)
        ET(L) = ETG(I)
        XT(L) = X(I)
        XX(L) = X(I)
 1040 CONTINUE
      IF (L .LE. 2) MEMG = 1
      IF (MEMG .EQ. 1) GO TO 1050
      WRITE (11,1010)
      CALL REG(L, XX, TT, RSULTG)
      IF (MEMTPG .EQ. 0) GO TO 1050
      NTAUKN = 0
      NTAUBN = 0
      WRITE (11,930)
      CALL TAUP(L, NTAUKG, NTAUBG, T, XT, ET, PBKG, TAUKG, PBBG, TPMAXG,
     1      TPMING)
C
CCCCCCC
C
C     CALCULATE DEPTHS , VELOCITIES AND ERRORS USING STANDARD FORMULAS
C
CCCCCCC
C
C
 1050 WRITE (11,1060)
 1060 FORMAT ('-------------------------------------------------------------------------',
     1        )
      IF (MEMM .EQ. 1 .AND. MEMN .EQ. 1) GO TO 1090
      IF (MEMM .EQ. 0 .AND. MEMN .EQ. 0) GO TO 1080
      IF (MEMM .EQ. 1 .AND. MEMN .EQ. 0) GO TO 1070
      CALL FMP
      GO TO 1090
 1070 CALL FN
      GO TO 1090
 1080 CALL FMPFN
```

```
      GO TO 1090
1090 IF (MEMC .EQ. 1) GO TO 1100
      CALL PCP
1100 IF (MEMG .EQ. 1) GO TO 1110
      CALL PG
1110 WRITE (7,1060)
      WRITE (11,1060)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     OUTPUT OF TRAVEL TIMES
C
C
CCCCCC
C
1120 IF (MEMN .EQ. 1) GO TO 1130
      ISIG = -1
      IF (VREDP .GT. 0.0) CALL REDUC(NMCG, VREDP, X, TN, ISIG)
1130 IF (MEMM .EQ. 1) GO TO 1140
      ISIG = -1
      IF (VREDP .GT. 0.0) CALL REDUC(NMCG, VREDP, X, TM, ISIG)
1140 IF (MEMC .EQ. 1) GO TO 1150
      ISIG = -1
      IF (VREDP .GT. 0.0) CALL REDUC(NMCG, VREDP, X, TC, ISIG)
1150 IF (MEMG .EQ. 1) GO TO 1160
      ISIG = -1
      IF (VREDP .GT. 0.0) CALL REDUC(NMCG, VREDP, X, TG, ISIG)
1160 CONTINUE
C
CCCCCCCCCCCCCCCCC
C
C
C     PLOTTING
C     ————————
C
CCCCCCCCCCCCCCCCC
C
      IF ((ABS(XMAX) + ABS(XMIN)) .NE. 0.) GO TO 1180
      XMAX = -999.0
      XMIN = 999.0
      DO 1170 I = 1, NMCG
        IF (X(I) .GT. XMAX) XMAX = X(I)
        IF (X(I) .LT. XMIN) XMIN = X(I)
1170 CONTINUE
1180 IF ((ABS(TMIN) + ABS(TMAX)) .NE. 0) GO TO 1230
      TMAX = -999.0
      TMIN = 999.0
      DO 1220 I = 1, NMCG
        IF (X(I) .GT. XMAX) XMAX = X(I)
        IF (X(I) .LT. XMIN) XMIN = X(I)
        IF (MEMN .EQ. 1) GO TO 1190
        IF (TN(I) .EQ. 0.0) GO TO 1190
        IF (TN(I) .GT. TMAX) TMAX = TN(I)
        IF (TN(I) .LT. TMIN) TMIN = TN(I)
1190    IF (MEMM .EQ. 1) GO TO 1200
        IF (TM(I) .EQ. 0.0) GO TO 1200
        IF (TM(I) .GT. TMAX) TMAX = TM(I)
        IF (TM(I) .LT. TMIN) TMIN = TM(I)
1200    IF (MEMC .EQ. 1) GO TO 1210
        IF (TC(I) .EQ. 0.0) GO TO 1210
        IF (TC(I) .GT. TMAX) TMAX = TC(I)
        IF (TC(I) .LT. TMIN) TMIN = TC(I)
1210    IF (MEMG .EQ. 1) GO TO 1220
        IF (TG(I) .EQ. 0.0) GO TO 1220
        IF (TG(I) .GT. TMAX) TMAX = TG(I)
        IF (TG(I) .LT. TMIN) TMIN = TG(I)
1220 CONTINUE
1230 IF(TMAX.EQ.-999.0.OR.TMIN.EQ.999.0) GO TO 1501
      IF(XMAX.EQ.-999.0.OR.XMIN.EQ.999.0) GO TO 1501
      XMAX = XMAX + 0.1 * (XMAX - XMIN)
      XMIN = XMIN - 0.1 * (XMAX - XMIN)
      TMAX = TMAX + 0.1 * (TMAX - TMIN)
      TMIN = TMIN - 0.1 * (TMAX - TMIN)
      IF (TSCALE .EQ. 0.0 .AND. DSCALE .EQ. 0.0) GO TO 1240
      TS = (TMAX - TMIN) * TSCALE
      DS = (XMAX - XMIN) * DSCALE
      GO TO 1250
C
C     PRODUCE A4 SIZE PLOT IF SCALING PARAMETRES=0
C
1240 DS = 7.0 * 2.54
      TS = 6.0 * 2.54
C
1250 IF (MEMRED .EQ. 0)CALL PAPER(1)
      IFRAME=200
      CALL GPSTOP(IFRAME)
      CALL FRAME
      CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(XMIN, XMAX, TMIN, TMAX)
      CALL CTRMAG(15)
      CALL SCALES
      CALL BORDER
C
C     PLOTTING PN
C
      IF (MEMN .EQ. 1) GO TO 1280
      DO 1260 I = 1, NMCG
        TIM(I) = 0.0
1260 XDIS(I) = 0.0
      L = 0
      DO 1270 I = 1, NMCG
        IF (TN(I) .EQ. 0.0) GO TO 1270
        L = L + 1
        TIM(L) = TN(I)
        XDIS(L) = X(I)
1270 CONTINUE
      CALL GRNPEN
      CALL PTPLOT(XDIS, TIM, 1, L, 45)
C
```

```
C     PLOTTING PMP
C
1280 IF (MEMM .EQ. 1) GO TO 1310
     DO 1290 I = 1, NMOG
       TIM(I) = 0.0
1290 XDIS(I) = 0.0
     L = 0
     DO 1300 I = 1, NMOG
       IF (TM(I) .EQ. 0.0) GO TO 1300
       L = L + 1
       TIM(L) = TM(I)
       XDIS(L) = X(I)
1300 CONTINUE
     CALL BLUPEN
     CALL PTPLOT(XDIS, TIM, 1, L, 43)
C
C     PLOTTING PCP
C
1310 IF (MEMC .EQ. 1) GO TO 1340
     DO 1320 I = 1, NMOG
       TIM(I) = 0.0
1320 XDIS(I) = 0.0
     L = 0
     DO 1330 I = 1, NMOG
       IF (TC(I) .EQ. 0.0) GO TO 1330
       L = L + 1
       TIM(L) = TC(I)
       XDIS(L) = X(I)
1330 CONTINUE
     CALL REDPEN
     CALL PTPLOT(XDIS, TIM, 1, L, 57)
C
C     PLOTTING PG
C
1340 IF (MEMG .EQ. 1) GO TO 1370
     DO 1350 I = 1, NMOG
       TIM(I) = 0.0
1350 XDIS(I) = 0.0
     L = 0
     DO 1360 I = 1, NMOG
       IF (TG(I) .EQ. 0.0) GO TO 1360
       L = L + 1
       TIM(L) = TG(I)
       XDIS(L) = X(I)
1360 CONTINUE
     CALL GRNPEN
     CALL PTPLOT(XDIS, TIM, 1, L, 45)
C
C     PLOTTING PN REGRESSION LINE
C
1370 IF (CDS .NE. 0.0) GO TO 1470
     IF (MEMN .EQ. 1) GO TO 1390
     XA(1) = X(MSTN)
     XA(2) = X(MENDN)
     COR1 = 0.0
     COR2 = 0.0
     IF (VREDP .LE. 0.0) GO TO 1380
     COR1 = XA(1) / VREDP
     COR2 = XA(2) / VREDP
1380 Y(1) = RSULTN(6) * XA(1) + RSULTN(7) - COR1
     Y(2) = RSULTN(6) * XA(2) + RSULTN(7) - COR2
     CALL GRNPEN
     CALL PTPLOT(XA, Y, 1, 2, -2)
C
C       PLOTTING PMP REGRESSION LINE
C
1390 IF (MEMM .EQ. 1) GO TO 1420
     L = 0
     DO 1410 I = MSTM, MENDM
       L = L + 1
       IF (VREDP .LE. 0.0) GO TO 1400
       COR2 = X(I) / VREDP
1400   XXA(L) = X(I)
       YY(L) = (RSULTM(6)*X(I)*X(I) + RSULTM(7)) ** 0.5 - COR2
1410 CONTINUE
     CALL BLUPEN
     CALL NSCURV(XXA, YY, 1, L)
C
C       PLOTTING PCP REGRESSION LINE
C
1420 IF (MEMC .EQ. 1) GO TO 1450
     L = 0
     DO 1440 I = MSTC, MENDC
       L = L + 1
       IF (VREDP .LE. 0.0) GO TO 1430
       COR2 = X(I) / VREDP
1430   XXA(L) = X(I)
       YY(L) = (RSULTC(6)*X(I)*X(I) + RSULTC(7)) ** 0.5 - COR2
1440 CONTINUE
     CALL REDPEN
     CALL NSCURV(XXA, YY, 1, L)
C
C     PLOTTING PG REGRESSION LINE
C
1450 IF (MEMG .EQ. 1) GO TO 1470
     XA(1) = X(MSTG)
     XA(2) = X(MENDG)
     COR1 = 0.0
     COR2 = 0.0
     IF (VREDP .LE. 0.0) GO TO 1460
     COR1 = XA(1) / VREDP
     COR2 = XA(2) / VREDP
1460 Y(1) = RSULTG(6) * XA(1) + RSULTG(7) - COR1
     Y(2) = RSULTG(6) * XA(2) + RSULTG(7) - COR2
     CALL GRNPEN
     CALL PTPLOT(XA, Y, 1, 2, -2)
C
1470 CALL BLKPEN
C
     CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
```

```
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL CTRMAG(10)
      CALL PLOTCS(4.0, 0.1, 'DISTANCE (KM)', 13)
      CALL CTRORI(1.0)
      CALL CTRMAG(10)
      CALL PLOTCS(0.15, 3.0, '(TIME-DIST/              ) SECS', 25)
      CALL PLOTNF(0.15, 5.0, VREDP, 1)
      CALL PSPACE(DS/25.4 + 0.1, DS/25.4 + 0.5, 0.1, TS/25.4 + 0.1)
      CALL CSPACE(DS/25.4 + 0.1, DS/25.4 + 0.5, 0.1, TS/25.4 + 0.1)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL BORDER
      CALL CTRORI(0.0)
      CALL CTRMAG(20)
      IF (ISTAT .NE. 0 .AND. ISHOT .EQ. 0) CALL PLOTCS(0.3, 9.0,
     1     'COMMON STATION ', 15)
      IF (ISTAT .EQ. 0 .AND. ISHOT .NE. 0) CALL PLOTCS(0.3, 9.0,
     1     'COMMON SHOT ', 12)
      IF (ISTAT .NE. 0 .AND. ISHOT .NE. 0) CALL PLOTCS(0.3, 9.0,
     1     'COMMON MID-POINT', 16)
      IF (CDS .NE. 0) CALL PLOTCS(0.3, 9.0, 'COMMON DISTANCE', 15)
      IF (ISTAT .NE. 0 .AND. ISHOT .EQ. 0) CALL PLOTCS(8.0, 9.0,
     1     ASTAT(ISTAT), 4)
      IF (ISTAT .EQ. 0 .AND. ISHOT .NE. 0) CALL PLOTCS(8.0, 9.0,
     1     ASHOT(ISHOT), 4)
      IF (ISTAT .NE. 0 .AND. ISHOT .NE. 0) CALL PLOTCS(8.0, 9.0,
     1     ASTAT(ISTAT), 4)
      IF (ISTAT .NE. 0 .AND. ISHOT .NE. 0) CALL PLOTCS(9.0, 9.0,
     1     ASHOT(ISHOT), 4)
      IF (CDS .NE. 0) CALL PLOTNF(8.0, 9.0, CDS, 1)
      CALL CTRMAG(8)
      CALL PLOTCS(1.0, 8.0, TITLE, 20)
      IF (MEMN .EQ. 1 .AND. MEMM .EQ. 1) GO TO 1480
      CALL PLOTCS(0.3, 7.5, 'AVERAGE CRUSTAL VEL', 20)
      CALL PLOTCS(0.3, 7.0, 'PN REFRACTOR VEL    ', 20)
      CALL PLOTCS(0.3, 6.5, 'DEPTH TO MOHO (PMP)', 20)
      CALL PLOTCS(0.3, 6.0, 'DEPTH TO MOHO (PN )', 20)
      CALL PLOTCS(0.3, 5.5, 'CRITICAL DIST IS    ', 20)
      CALL PLOTNF(6.0, 7.5, VAV, 2)
      CALL PLOTNF(6.0, 7.0, VN, 2)
      CALL PLOTNF(6.0, 6.5, ZM, 2)
      CALL PLOTNF(6.0, 6.0, ZN, 2)
      CALL PLOTNF(6.0, 5.5, XCRIT, 2)
      CALL PLOTCS(7.0, 7.5, '+/-', 3)
      CALL PLOTCS(7.0, 7.0, '+/-', 3)
      CALL PLOTCS(7.0, 6.5, '+/-', 3)
      CALL PLOTCS(7.0, 6.0, '+/-', 3)
      CALL PLOTCS(7.0, 5.5, '+/-', 3)
      CALL PLOTNF(8.4, 7.5, EVAV, 4)
      CALL PLOTNF(8.4, 7.0, EVN, 4)
      CALL PLOTNF(8.4, 6.5, EZM, 4)
      CALL PLOTNF(8.4, 6.0, EZN, 4)
      CALL PLOTNF(8.4, 5.5, EXCRIT, 4)
C
1480 IF (MEMC .EQ. 1) GO TO 1490
      CALL PLOTCS(0.1, 4.0, 'AVERAGE VEL TO PCP  ', 20)
      CALL PLOTCS(0.1, 3.5, 'DEPTH TO PCP        ', 20)
      CALL PLOTNF(6.0, 4.0, VC, 2)
      CALL PLOTNF(6.0, 3.5, ZC, 2)
      CALL PLOTCS(7.0, 4.0, '+/-', 3)
      CALL PLOTCS(7.0, 3.5, '+/-', 3)
      CALL PLOTNF(8.4, 4.0, EVC, 4)
      CALL PLOTNF(8.4, 3.5, EZC, 4)
C
1490 IF (MEMG .EQ. 1) GO TO 1500
      CALL PLOTCS(0.1, 2.0, 'VELOCITY OF PG IS   ', 20)
      CALL PLOTCS(0.1, 1.5, 'INTERCEPT OF PG IS  ', 20)
      CALL PLOTNF(6.0, 2.0, VG, 2)
      CALL PLOTNF(6.0, 1.5, ZG, 2)
      CALL PLOTCS(7.0, 2.0, '+/-', 3)
      CALL PLOTCS(7.0, 1.5, '+/-', 3)
      CALL PLOTNF(8.4, 2.0, EVG, 4)
      CALL PLOTNF(8.4, 1.5, EZG, 4)
1500 CALL TIME(6, 0, ITIME)
      CALL PLOTCS(2.0, 1.0, ITIME, 20)
      DO 1503 I=1,180
        IF(ISTAT.NE.0.AND.ASTAT(ISTAT).EQ.KSTH(I)) XM25=DIST25(I)
        IF(ISHOT.NE.0.AND.ASHOT(ISHOT).EQ.KSTH(I)) XM25=DIST25(I)
1503   CONTINUE
      IF(MEMC.EQ.1) VC=0.0
      IF(MEMC.EQ.1) ZC=0.0
      IF(MEMM.EQ.1) VM=0.0
      IF(MEMM.EQ.1) ZM=0.0
      IF(MEMN.EQ.1) VC=0.0
      IF(MEMN.EQ.1) ZC=0.0
      IF(ISTAT.NE.0) WRITE(12,1504) XM25,VAV,ZM,VN,ZN,VC,ZC,ASTAT(ISTAT)
      IF(ISHOT.NE.0) WRITE(12,1504) XM25,VAV,ZM,VN,ZN,VC,ZC,ASHOT(ISHOT)
1504 FORMAT(7F10.3,4X,A4)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     PLOTTING TAUP RESULTS
C     ---------------------
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
1501 IF ((MEMTPN + MEMTPM + MEMTPC + MEMTPG) .EQ. 0) GO TO 1690
      PMAX = -999.0
      PMIN = 999.0
      TMAX = -999.0
      TMIN = 999.0
      IF (MEMTPN .EQ. 0) GO TO 1530
      DO 1510 I = 1, NTAUKN
        IF (PBKN(I) .GT. PMAX) PMAX = PBKN(I)
        IF (PBKN(I) .LT. PMIN) PMIN = PBKN(I)
        IF (TAUKN(I) .GT. TMAX) TMAX = TAUKN(I)
        IF (TAUKN(I) .LT. TMIN) TMIN = TAUKN(I)
1510 CONTINUE
      DO 1520 I = 1, NTAUBN
```

```
      IF (PBBN(I) .GT. PMAX) PMAX = PBBN(I)
      IF (PBBN(I) .LT. PMIN) PMIN = PBBN(I)
      IF (TPMINN(I) .GT. TMAX) TMAX = TPMINN(I)
      IF (TPMINN(I) .LT. TMIN) TMIN = TPMINN(I)
      IF (TPMAXN(I) .GT. TMAX) TMAX = TPMAXN(I)
      IF (TPMAXN(I) .LT. TMIN) TMIN = TPMAXN(I)
1520 CONTINUE
1530 IF (MEMTPM .EQ. 0) GO TO 1560
     DO 1540 I = 1, NTAUKM
      IF (PBKM(I) .GT. PMAX) PMAX = PBKM(I)
      IF (PBKM(I) .LT. PMIN) PMIN = PBKM(I)
      IF (TAUKM(I) .GT. TMAX) TMAX = TAUKM(I)
      IF (TAUKM(I) .LT. TMIN) TMIN = TAUKM(I)
1540 CONTINUE
     DO 1550 I = 1, NTAUBM
      IF (PBBM(I) .GT. PMAX) PMAX = PBBM(I)
      IF (PBBM(I) .LT. PMIN) PMIN = PBBM(I)
      IF (TPMINM(I) .GT. TMAX) TMAX = TPMINM(I)
      IF (TPMINM(I) .LT. TMIN) TMIN = TPMINM(I)
      IF (TPMAXM(I) .GT. TMAX) TMAX = TPMAXM(I)
      IF (TPMAXM(I) .LT. TMIN) TMIN = TPMAXM(I)
1550 CONTINUE
1560 IF (MEMTPC .EQ. 0) GO TO 1590
     DO 1570 I = 1, NTAUKC
      IF (PBKC(I) .GT. PMAX) PMAX = PBKC(I)
      IF (PBKC(I) .LT. PMIN) PMIN = PBKC(I)
      IF (TAUKC(I) .GT. TMAX) TMAX = TAUKC(I)
      IF (TAUKC(I) .LT. TMIN) TMIN = TAUKC(I)
1570 CONTINUE
     DO 1580 I = 1, NTAUBC
      IF (PBBC(I) .GT. PMAX) PMAX = PBBC(I)
      IF (PBBC(I) .LT. PMIN) PMIN = PBBC(I)
      IF (TPMINC(I) .GT. TMAX) TMAX = TPMINC(I)
      IF (TPMINC(I) .LT. TMIN) TMIN = TPMINC(I)
      IF (TPMAXC(I) .GT. TMAX) TMAX = TPMAXC(I)
      IF (TPMAXC(I) .LT. TMIN) TMIN = TPMAXC(I)
1580 CONTINUE
1590 IF (MEMTPG  EQ. 0) GO TO 1620
     DO 1600 I = 1, NTAUKG
      IF (PBKG(I) .GT. PMAX) PMAX = PBKG(I)
      IF (PBKG(I) .LT. PMIN) PMIN = PBKG(I)
      IF (TAUKG(I) .GT. TMAX) TMAX = TAUKG(I)
      IF (TAUKG(I) .LT. TMIN) TMIN = TAUKG(I)
1600 CONTINUE
     DO 1610 I = 1, NTAUBG
      IF (PBBG(I) .GT. PMAX) PMAX = PBBG(I)
      IF (PBBG(I) .LT. PMIN) PMIN = PBBG(I)
      IF (TPMING(I) .GT. TMAX) TMAX = TPMING(I)
      IF (TPMING(I) .LT. TMIN) TMIN = TPMING(I)
      IF (TPMAXG(I) .GT. TMAX) TMAX = TPMAXG(I)
      IF (TPMAXG(I) .LT. TMIN) TMIN = TPMAXG(I)
1610 CONTINUE
1620 PMAX = PMAX + 0.1 * (PMAX - PMIN)
     PMIN = PMIN - 0.1 * (PMAX - PMIN)
     TMAX = TMAX + 0.1 * (TMAX - TMIN)
     TMIN = TMIN - 0.1 * (TMAX - TMIN)
     IF (PSCALE .EQ. 0.0 .AND. TASCAL .EQ. 0.0) GO TO 1630
     TS = (TMAX - TMIN) * TASCAL
     PS = (PMAX - PMIN) * PSCALE
     GO TO 1640
C
C     PRODUCE A4 SIZE PLOT IF SCALING PARAMETRES=0
C
1630 PS = 7.0 * 2.54
     TS = 6.0 * 2.54
C
1640 CALL FRAME
     CALL PSPACE(0.1, PS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(0.0, PS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     CALL MAP(PMIN, PMAX, TMIN, TMAX)
     CALL CTRMAG(15)
     CALL BLKPEN
     CALL SCALES
     CALL BORDER
C
     IF (MEMTPN .EQ. 0) GO TO 1650
     CALL GRNPEN
     CALL PTPLOT(PBKN, TAUKN, 1, NTAUKN, 45)
     CALL PTPLOT(PBBN, TPMINN, 1, NTAUBN, -2)
     CALL PTPLOT(PBBN, TPMAXN, 1, NTAUBN, -2)
1650 IF (MEMTPM .EQ. 0) GO TO 1660
     CALL BLUPEN
     CALL PTPLOT(PBKM, TAUKM, 1, NTAUKM, 43)
     CALL PTPLOT(PBBM, TPMINM, 1, NTAUBM, -2)
     CALL PTPLOT(PBBM, TPMAXM, 1, NTAUBM, -2)
1660 IF (MEMTPC .EQ. 0) GO TO 1670
     CALL REDPEN
     CALL PTPLOT(PBKC, TAUKC, 1, NTAUKC, 57)
     CALL PTPLOT(PBBC, TPMINC, 1, NTAUBC, -2)
     CALL PTPLOT(PBBC, TPMAXC, 1, NTAUBC, -2)
1670 IF (MEMTPG .EQ. 0) GO TO 1680
     CALL REDPEN
     CALL PTPLOT(PBKG, TAUKG, 1, NTAUKG, 57)
     CALL PTPLOT(PBBG, TPMING, 1, NTAUBG, -2)
     CALL PTPLOT(PBBG, TPMAXG, 1, NTAUBG, -2)
1680 CALL BLKPEN
C
C     ANNOTATION
C
     CALL PSPACE(0.0, PS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     CALL CSPACE(0.0, PS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     CALL MAP(0.0, 10.0, 0.0, 10.0)
     CALL CTRMAG(10)
     CALL PLOTCS(4.0, 0.5, 'P (S/KM)', 8)
     CALL CTRORI(1.0)
     CALL CTRMAG(10)
     CALL PLOTCS(0.15, 3.0, 'TAU (S)', 7)
     CALL PSPACE(PS/25.4 + 0.1, PS/25.4 + 0.5, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(PS/25.4 + 0.1, PS/25.4 + 0.5, 0.1, TS/25.4 + 0.1)
     CALL MAP(0.0, 5.0, 0.0, 10.0)
```

```
      CALL BORDER
      CALL CTRORI(0.0)
      CALL CTRMAG(20)
      CALL PLOTCS(0.5, 9.0, TITLE, 20)
C
      GO TO 1690
C
C      ERROR MESSAGES
C      --------------
1690 WRITE (6,1700)
1700 FORMAT (/'DO YOU WANT TO CONTINUE WITH ANOTHER SECTION? Y=1')
      READ (5,1710) MEMRED
1710 FORMAT (I1)
      IF (MEMRED .EQ. 1) GO TO 10
      CALL GREND
      STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REG(L, X, T, RSULT)
C     ------------------------------
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*8 T(200), X(200), RSULT(20)
      DIMENSION TVALUE(31)
      DATA TVALUE /12.71, 4.30, 3.18, 2.78, 2.57, 2.45, 2.36, 2.31,
     1     2.26, 2.23, 2.20, 2.18, 2.16, 2.15, 2.13, 2.12, 2.11, 2.10,
     2     2.09, 2.09, 2.08, 2.07, 2.07, 2.06, 2.06, 2.05, 2.05, 2.05,
     3     2.04, 2.04, 2.02/
C
C
      IFAIL = 0
      CALL G02CAF(L, X, T, RSULT, IFAIL)
      IF (IFAIL .EQ. 1) GO TO 50
      IF (IFAIL .EQ. 2) GO TO 70
C      WRITE (7,10) (RSULT(I),I=1,5)
C  10 FORMAT (///'MEAN OF INDEPENDANT VARIABLE  = ', F15.5, /
C    1        'MEAN OF DEPENDANT VARIABLE = ', F15.5, /
C    2        'STD DEV OF IND VAR = ', F15.5, /'STD DEV OF DEP VAR = ',
C    3        F15.5, /'CORR COEF = ', F15.5)
C
C      WRITE (7,20) RSULT(6), RSULT(8), RSULT(10)
C  20 FORMAT (///'REGRESSION COEFF = ', F15.5, /'STD ERR OF COEFF = ',
C    1        F15.5, /'T-VALUE FOR COEF = ', F15.5)
C
C      WRITE (7,30) RSULT(7), RSULT(9), RSULT(11)
C  30 FORMAT (///'REGESSION CONST = ', F15.5, /'STD ERR OF CONST = ',
C    1        F15.5, /'T-VALUE FOR CONST = ', F15.5)
C
C
C      WRITE (7,40) (RSULT(I),I=12,20)
C  40 FORMAT (///'                  ANALYSIS OF RERESSION TABLE', //
C    1
C    2'  SOURCE     SUM OF SQUARES   D.F.    MEAN SQUARE    F-VALUE', //
C    3       'DUE TO REGR', F15.5, F8.0, 2F15.5, /'ABOUT REGR ', F15.5,
C    4       F8.0, F15.5, /'TOTAL       ', F15.5, F8.0, ///)
      NDF = RSULT(20)
      IF (NDF .GT. 30.0) NDF = 31.0
      RSULT(8) = RSULT(8) * TVALUE(NDF)
      RSULT(9) = RSULT(9) * TVALUE(NDF)
C
      GO TO 90
C
C      ERRORS
C      ------
C
  50 WRITE (11,60) IFAIL
  60 FORMAT (//'IFAIL = ', I2, ' NO OF SAMPLES (N) < 2')
      GO TO 90
  70 WRITE (11,80) IFAIL
  80 FORMAT (//'IFAIL = ', I2, ' ALL X (OR Y) ARE IDENTICAL')
      GO TO 90
C
  90 RETURN
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE REDUC(N, VRED, X, T, ISIG)
C     -------------------------------------
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      DIMENSION T(200), X(200), T6(200), T8(200), ASHOT(110),
     1          ASTAT(110), IST(110), ISH(110)
      COMMON /RED/ ASTAT, ASHOT, ISTAT, ISHOT, IST, ISH
C
      IF (ISIG .GT. 1) GO TO 20
      DO 10 I = 1, N
        IF (T(I) .EQ. 0.0) GO TO 10
        T(I) = T(I) + ISIG * X(I) / VRED
  10 CONTINUE
      GO TO 160
  20 DO 30 I = 1, N
        IF (T(I) .EQ. 0.0) GO TO 30
        T6(I) = T(I) - X(I) / 6.0
        T8(I) = T(I) - X(I) / 8.0
  30 CONTINUE
      IF (ISIG .EQ. 3) GO TO 40
      IF (ISIG .EQ. 4) GO TO 50
      IF (ISIG .EQ. 5) GO TO 60
      WRITE (11,110)
```

```
      GO TO 70
   40 WRITE (11,120)
      GO TO 70
   50 WRITE (11,130)
      GO TO 70
   60 WRITE (11,140)
      GO TO 70
   70 DO 100 I = 1, N
        IF (T(I) .EQ. 0.0) GO TO 100
   80   M = IST(I)
        J = ISH(I)
        GO TO 90
   90   WRITE (11,150) ASTAT(M), ASHOT(J), X(I), T(I), T6(I), T8(I)
  100 CONTINUE
  110 FORMAT (///, 9X, '  DIST      TIME TO PN   VRED=6.0  VRED=8.0 '/)
  120 FORMAT (///, 9X, '  DIST      TIME TO PMP  VRED=6.0  VRED=8.0 '/)
  130 FORMAT (///, 9X, '  DIST      TIME TO PCP  VRED=6.0  VRED=8.0 '/)
  140 FORMAT (///, 9X, '  DIST      TIME TO PG   VRED=6.0  VRED=8.0 '/)
  150 FORMAT (A4, 1X, A4, 4F10.4)
      GO TO 160
  160 RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE PMPPN
C     ________________
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       CALCULATES VELOCITIES AND DEPTHS USING REGRESSION
C       RESULTS AND STANDARD FORMULAS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*8 RSULTN(20), RSULTM(20)
      DIMENSION TITLES(20), TVALUE(31)
      COMMON /P/ VAV, EVAV, VN, EVN, ZM, EZM, ZN, EZN, XCRIT, EXCRIT,
     1       RSULTM, RSULTN
C
C
C
C
C
      VAV = 1 / RSULTM(6) ** 0.5
      EVAV = RSULTM(8) / (2.0*RSULTM(6)**1.5)
      VN = 1 / RSULTN(6)
      EVN = RSULTN(8) / (RSULTN(6)**2)
      IF ((VN**2 - VAV**2) .LT. 0.0) GO TO 50
      IF (RSULTM(7) .GE. 0.0) GO TO 10
      RSULTM(7) = -RSULTM(7)
      ZM = -VAV * (RSULTM(7)**0.5) / 2
      GO TO 20
   10 ZM = VAV * (RSULTM(7)**0.5) / 2
   20 EZM = ZM * (RSULTM(8)/VAV + RSULTM(9)/(2*RSULTM(7)))
      ZN = RSULTN(7) * VN * VAV / (2*(VN**2 - VAV**2)**0.5)
      EZN = ZN * (RSULTN(9)/RSULTN(7) + EVN/VN + EVAV/VAV + (VN*EVN +
     1VAV*EVAV)/(VN**2 - VAV**2))
      XCRIT = ZM * VAV / (VN**2 - VAV**2) ** 0.5
      EXCRIT = XCRIT * (EZM/ZM + EVAV/VAV + (VN*EVN + VAV*EVAV)/(VN**2 -
     1 VAV**2))
      WRITE (7,30) VAV, EVAV, ZM, EZM
   30 FORMAT (///'————— PMP WAR RESULTS —————'/
     1        'AVERAGE CRUSTAL VELOCITY IS', F10.4, '+/-', F15.8, /
     2        'DEPTH TO MOHO IS ', F10.4, '+/-', F13.8, /)
      WRITE (7,40) VN, EVN, ZN, EZN, XCRIT, EXCRIT
   40 FORMAT (///'————— PN REF RESULTS —————'/
     1        'PN VELOCITY IS', F10.4, '+/-', F13.8, /
     2        'DEPTH TO MOHO (USING AV. CRUSTAL VEL FROM PMP) IS', F10.4,
     3        '+/-', F13.8, /'CRITICAL DISTANCE IS ', F10.4, '+/-',
     4        F13.8, /)
      GO TO 70
C
C      ERRORS
C
   50 WRITE (6,60)
   60 FORMAT (//'VN**2-VAV**2  IS LESS THAN ZERO')
      GO TO 70
C
C      RETURN
C
   70 RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE PN
C     _____________
C
C
CCCC
C
C
C
      REAL*8 RSULTN(20), RSULTM(20)
      DIMENSION TITLES(20), TVALUE(31)
      COMMON /P/ VAV, EVAV, VN, EVN, ZM, EZM, ZN, EZN, XCRIT, EXCRIT,
     1       RSULTM, RSULTN
C
C
C
C
C
      VAV = 6.5000
      EVAV = 0.0
```

```
      VN = 1 / RSULTN(6)
      EVN = RSULTN(8) / (RSULTN(6)**2)
      IF ((VN**2 - VAV**2) .LT. 0.0) GO TO 20
      ZM = 0.0
      EZM = 0.0
      ZN = RSULTN(7) * VN * VAV / (2*(VN**2 - VAV**2)**0.5)
      EZN = ZN * (RSULTN(9)/RSULTN(7) + EVN/VN + EVAV/VAV + (VN*EVN +
     1VAV*EVAV)/(VN**2 - VAV**2))
      XCRIT = ZN * VAV / (VN**2 - VAV**2) ** 0.5
      EXCRIT = XCRIT * (EZN/ZN + EVAV/VAV + (VN*EVN + VAV*EVAV)/(VN**2 -
     1 VAV**2))
      WRITE (7,10) VN, EVN, ZN, EZN, XCRIT, EXCRIT
   10 FORMAT (//'************** PN REF RESULTS **********'/'PN VEL IS ',
     1       F10.4, '+/-', F13.8, /
     2       'DEPTH TO MOHO (USING AV. CRUSTAL VEL=6.5)      IS', F10.4,
     3       '+/-', F13.8, /'CRITICAL DISTANCE IS ', F10.4, '+/-',
     4       F13.8,  /)
      GO TO 40
C
C      ERRORS
C
   20 WRITE (6,30)
   30 FORMAT (//'VN**2-VAV**2  IS LESS THAN ZERO')
      GO TO 40
C
C      RETURN
C
   40 RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FMP
C      ----------------
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       CALCULATES VELOCITIES AND DEPTHS USING REGRESSION
C       RESULTS AND STANDARD FORMULAS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      REAL*8 RSULTN(20), RSULTM(20)
      DIMENSION TITLES(20), TVALUE(31)
      COMMON /P/ VAV, EVAV, VN, EVN, ZM, EZM, ZN, EZN, XCRIT, EXCRIT,
     1       RSULTM, RSULTN
C
      VAV = 1 / RSULTM(6) ** 0.5
      EVAV = RSULTM(8) / (2.0*RSULTM(6)**1.5)
      IF (RSULTM(7) .GE. 0.0) GO TO 10
      RSULTM(7) = -RSULTM(7)
      ZM = -VAV * (RSULTM(7)**0.5) / 2
      GO TO 20
   10 ZM = VAV * (RSULTM(7)**0.5) / 2
   20 EZM = ZM * (RSULTM(8)/VAV + RSULTM(9)/(2*RSULTM(7)))
      WRITE (7,30) VAV, EVAV, ZM, EZM
   30 FORMAT (///'---------- FMP WAR RESULTS -----------'/
     1       'AVERAGE CRUSTAL VELOCITY ABOVE MOHO IS', F10.4, '+/-',
     2       F13.8, /'DEPTH TO MOHO IS', F10.4, '+/-', F13.8, /)
      VN = 0.0
      EVN = 0.0
      ZN = 0.0
      EZN = 0.0
C
      GO TO 40
C
C      ERRORS
C
C
C      RETURN
C
   40 RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE PCP
C      ----------------
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       CALCULATES VELOCITIES AND DEPTHS USING REGRESSION
C       RESULTS AND STANDARD FORMULAS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      REAL*8 RSULTC(20)
      DIMENSION TITLES(20), TVALUE(31)
      COMMON /PC/ VC, EVC, ZC, EZC, RSULTC
C
      VC = 1 / RSULTC(6) ** 0.5
      EVC = RSULTC(8) / (2.0*RSULTC(6)**1.5)
      IF (RSULTC(7) .GE. 0.0) GO TO 10
      RSULTC(7) = -RSULTC(7)
      ZC = -VC * (RSULTC(7)**0.5) / 2
      GO TO 20
   10 ZC = VC * (RSULTC(7)**0.5) / 2
   20 EZC = ZC * (RSULTC(8)/VC + RSULTC(9)/(2*RSULTC(7)))
      WRITE (7,30) VC, EVC, ZC, EZC
   30 FORMAT (///'---------- PCP WAR RESULTS -----------'/
     1       'AVERAGE CRUSTAL VELOCITY ABOVE CONRAD IS', F10.4, '+/-',
```

```
     2       F13.8, /'DEPTH TO CONRAD IS', F10.4, '+/-', F13.8, /)
C
      GO TO 40
C
C      ERRORS
C
C
C      RETURN
C
   40 RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE PG
C
C
CCCC
C
C
C
      REAL*8 RSULTG(20)
      DIMENSION TITLES(20), TVALUE(31)
      COMMON /G/ VG, EVG, ZG, EZG, RSULTG
C
C
C
      VG = 1 / RSULTG(6)
      EVG = RSULTG(8) / (RSULTG(6)**2)
      ZG = 0.0
      EZG = 0.0
      VAV = 4.8
      IF (VG .LT. VAV) VAV = VG / 2.0
      ZG = RSULTG(7) * VG * VAV / (2*(VG**2 - VAV**2)**0.5)
      EZG = ZG * (RSULTG(9)/RSULTG(7) + EVG/VG + (VG*EVG)/(VG**2 - VAV**
     12))
      WRITE (7,10) VG, EVG, VAV, ZG, EZG
   10 FORMAT (//'************** PG REF RESULTS ********'//'PG VEL IS ',
     1       F10.4, '+/-', F13.8, /
     2       'DEPTH TO BASEMENT USING CRUSTAL VEL=', F5.2, ' IS ',
     3       F10.4, '+/-', F13.8)
C
C      RETURN
C
   20 RETURN
      END
C
```

```
C
CA    AHJ Lewis
C*****************************************************************
C*                                                               *
C*                  PROGRAMME  TTCORR                            *
C*                                                               *
C*                                                               *
C*  AHJ LEWIS       Dept. of Geological Sciences          NOV,1985 *
C*****************************************************************
C
C
C                        DESCRIPTION
C
C
C     This programme corrects the travel-times of any phase held in a
C     travel-time matrix for shallow structure. The apparent surface
C     velocity is calculated along common station lines to find the
C     corrections for the shots, and, along common shot lines for the
C     station.
C
C     The output is in a form that is easily adjusted for input to the
C     programme  TTERMWAR.U  which inverts the travel-times of
C     a wide-angle reflection phase to depths and average velocities.
C     The corrected values for pn can also be input to the time-term
C     TTERMN.U.
C
C
C     INPUT
C
C          UNIT 4  - travle-time matrix assigned internally
C          UNIT 5  - default to screen
C          UNIT 8  - distance matrix assigned internally
C          UNIT 10 - shallow structure details : example input held in
C                    TTCORR.IN
C
C     OUTPUT
C
C          UNIT 3  - corrected travel-times. Example held in TTCORR-PMP
C          UNIT 6  - screen
C          UNIT 7  - details of correction apparent velocities etc..
C
C
C ---------------------------------------------------------------------
C
C     SUBROUTINES CALLED :  MATRD    - Reads a travel-time matrix
C                           SHALLSH  - calculates apparent velocities
C                                      and corrections along CST and CSH
C                                      lines
C                           MATCOR   - outputs the corrected data in the
C                                      form required by *SURFACE2 which
C                                      can be used to plot the tt-matrices
C
C
C---------------------------------------------------------------------
CD-
```

```
      DIMENSION KSTH(180), DIST25(180), TMAT(110,110), XMAT(110,110),
     1          ET(110,110), ASTAT(110), ASHOT(110), TTRMPG(200),
     2          TITLES(20), KSITE(200), VG(200), VAVPG(200),
     3          ASH(110,110), TSH(110,110), XSH(110,110), TST(110,110),
     4          XST(110,110), TSHC(110,110), XSHC(110,110),
     5          XSTC(110,110), AST(110,110), DSH(2000), DST(2000),
     6          TSTC(110,110), NST(110), NSH(110), TT(2000), XX(2000)
      INTEGER ASTAT, ASHOT, BLANK, AST, ASH
      DATA M25B /'M25B'/



C
C
C******************** READ IN INPUT DATA *******************
C
C                    read in data from matrices
C                     read in data from unit 5
C               read in distances fom m25b from unit 10
C

      READ (5,270) TITLE
      WRITE (6,270) TITLE
      WRITE (7,270) TITLE
      WRITE (3,270) TITLE
      CALL MATRD(ASTAT, ASHOT, TMAT, ET, XMAT)
      READ (5,10) NUMC, VADEF, HD , IC
      WRITE (7,10) NUMC, VADEF, HD ,IC
      WRITE (3,10) NUMC, VADEF, HD ,IC
   10 FORMAT (I5, 2F5.2,  I5)
      DO 20 I = 1, NUMC
        READ (5,300) KSITE(I), VG(I), VAVPG(I), TTRMPG(I)
   20 WRITE (7,300) KSITE(I), VG(I), VAVPG(I), TTRMPG(I)
   30 CALL FTNOMD('ASSIGN 10=GPT9:XMAT-M25B', 24)
      READ (10,270) TITLES
      READ (10,280) (KSTH(I),I=1,180)
      READ (10,290) (DIST25(I),I=1,180)



C
C
C********************* LOOP OVER CSH ***************************
C*********************      & CST     ***************************
C
C


C ---------- remove zero tt along csh lines and copy non-zero
C              locations to new arays for input to SHALLTH

      L = 0
      INU = 0
      WRITE (6,40)
   40 FORMAT (1X, 'AT DO LOOP 40')
      DO 70 ICSH = 1, 71
        L = 0
        DO 50 J = 1, 110
          IF (TMAT(J,ICSH) .EQ. 0.0) GO TO 50
          L = L + 1
          ASH(L,ICSH) = ASTAT(J)
          TSH(L,ICSH) = TMAT(J,ICSH)
          XSH(L,ICSH) = XMAT(J,ICSH)
   50   CONTINUE
        NSH(ICSH) = L
   70 CONTINUE


C ---------- remove zero tt along csh lines and copy non-zero
C              locations to new arays for input to SHALLTH

      L = 0
      DO 100 ICST = 1, 110
        L = 0
        DO 90 J = 1, 71
          IF (TMAT(ICST,J) .EQ. 0.0) GO TO 90
          L = L + 1
          AST(L,ICST) = ASHOT(J)
          TST(L,ICST) = TMAT(ICST,J)
          XST(L,ICST) = XMAT(ICST,J)
   90   CONTINUE
        NST(ICST) = L
  100 CONTINUE


C-------- call SHALLTH and correct for shallow structure
C-------- along CSH and CST directions

      WRITE (6,110)
  110 FORMAT (1X, 'ENTERING SHALL')
      NUM = 71
      CALL SHALL(ASH, TSH, XSH, NSH, TSHC, XSHC, KSITE, VG, VAVPG,
     1      TTRMPG, NUMC, VADEF, ASTAT, ASHOT, NUM, HD,IC)

      NUM = 110
      CALL SHALL(AST, TST, XST, NST, TSTC, XSTC, KSITE, VG, VAVPG,
     1      TTRMPG, NUMC, VADEF, ASTAT, ASHOT, NUM, HD,IC)



C------- looping over cst directions correct travel-times for
C------- shots and stations and correct the station/shot offsets
C------- and all site distances from M25B
```

```
C
      DO 150 I = 1, 110
        L = 0
        DO 150 J = 1, 71

C check tt is not zero

          IF (TMAT(I,J) .EQ. 0.0) GO TO 150
          L = L + 1

C find locations of station and shot in M25B distance array

          DO 120 JJ = 1, 180
            IF (ASTAT(I) .EQ. KSTH(JJ)) IST = JJ
            IF (ASHOT(J) .EQ. KSTH(JJ)) ISH = JJ
 120      CONTINUE

C find position of station corrections in shot arrays

          NN = NSH(J)
          DO 130 JH = 1, NN
            IF (ASTAT(I) .EQ. ASH(JH,J)) JJ = JH
 130      CONTINUE

C correct tt

          T = TMAT(I,J) - TSTC(L,I) - TSHC(JJ,J)

C check which direction to correct distances from M25B

          IF (DIST25(IST).GT.DIST25(ISH))
     .    XST25 = DIST25(IST) - XSTC(L,I)
          IF (DIST25(IST).GT.DIST25(ISH))
     .    XSH25 = DIST25(ISH) + XSHC(JJ,J)
          IF (DIST25(ISH).GT.DIST25(IST))
     .    XST25 = DIST25(IST) + XSTC(L,I)
          IF (DIST25(ISH).GT.DIST25(IST))
          XSH25 = DIST25(ISH) - XSHC(JJ,J)

C  calculate new station/shot offset

          X = ABS(XST25 - XSH25)

C output results to unit 3

          WRITE (3,160) XST25, XSH25, T, X, ASTAT(I), ASH(JJ,J),
     1    ASHOT(J), AST(L,I)

C put results into arrays for call to matcor

          INJ = INJ + 1
          DST(INJ) = XST25
          DSH(INJ) = XSH25
          TT(INJ)  = T
          XX(INJ)  = X
C now do next CST line

 150 CONTINUE


C------ CALL MATCOR - OUTPUTS *SURFACE2 SOURCE AND DATA FILES

      CALL MATCOR(DST, DSH, TT, XX, INJ)

C************************ FORMAT STATEMENTS ************************
C
 160 FORMAT(4F10.3, 4(1X,A4))
 180 FORMAT (1X, A4, 1X, A4, 6F10.3)
 200 FORMAT (1X, A4, 1X, A4, 4F10.3)
 210 FORMAT (I2)
 220 FORMAT (20A4)
 230 FORMAT (1X, A4, '/', A4, F10.3)
 250 FORMAT (I5)
 260 FORMAT ((20A4))
 270 FORMAT (20A4)
 280 FORMAT (3(60A4))
 290 FORMAT (3(60F10.4))
 300 FORMAT ((A4,1X,3F10.3))
 330 FORMAT (1X, A4, '/', A4, 2F10.3)
 350 FORMAT (110A4)
 360 FORMAT (71(F10.4,F10.4))
C------------------------------------------------------------------

C corrections completed

      STOP
      END

C
C
C
C********************** SUBROUTINE MATREAD ************************
C
C
C
      SUBROUTINE MATRD(ASTAT, ASHOT, TMAT, ET, XMAT)
      DIMENSION TMAT(110,110), ET(110,110),
     1          ASTAT(110), ASHOT(110), XMAT(110,110), XSTAT(110),
     2          XSHOT(110)
      INTEGER XSHOT, XSTAT, ASTAT, ASHOT
      DATA PG /'PG  '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/

   10 WRITE (6,20)
   20 FORMAT (/'TRAVEL TIMES FOR WHICH PHASE? (EG. PG,PC1,PCP,PMP,PN)')
      READ (5,200) P
```

```
      WRITE (6,200) P
      IF (P .EQ. PG) GO TO 30
      IF (P .EQ. PCP) GO TO 40
      IF (P .EQ. PMP) GO TO 50
      IF (P .EQ. PN) GO TO 60
      IF (P .EQ. PN1) GO TO 70
   30 CALL FTNOMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 80
   40 CALL FTNOMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 80
   50 CALL FTNOMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 80
   60 CALL FTNOMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 80
   70 CALL FTNOMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 80
   80 CONTINUE
      CALL FTNOMD('ASSIGN 8=GPT9:XMAT-CSSP', 23)
C
C      CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
C      IF NOT STOP
C
   90 READ (4,200) TIN
      IF (TIN .EQ. P) GO TO 110
      WRITE (6,100) TIN, P
  100 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
      STOP
C
C      READ IN VALUES ALREADY IN MATRIX FILE
C
  110 WRITE (6,190)
      READ (4,160) (ASTAT(I),I=1,110)
      READ (4,160) (ASHOT(I),I=1,71)
      READ (8,200) ADUMP
      READ (8,160) (XSTAT(I),I=1,110)
      READ (8,160) (XSHOT(I),I=1,71)
  120 DO 140 I = 1, 110
        READ (4,170) (TMAT(I,J),ET(I,J),J=1,71)
        READ (8,170) (XMAT(I,J),J=1,71)
  130   IF (ASTAT(I) .EQ. XSTAT(I)) GO TO 140
        WRITE (6,180)
        STOP
  140 CONTINUE
  150 CONTINUE
  160 FORMAT (110A4)
  170 FORMAT (71(F10.4,F10.4))
  180 FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
  190 FORMAT (/'READING DATA FROM MATRIX FILE')
  200 FORMAT (A4)
      RETURN
      END


C
C*********************************************************************
C
C
C
C*********************************************************************
C
C
C ********************  SHALL  ******************************
C
C
C*********************************************************************
C
C
      SUBROUTINE SHALL(ALPHA1, TSC, XSC, NS, TCOR, XCOR, KSITE, VPG,
     1           VAVPG, TTRMPG, NUMC, VADEF, ASTAT, ASHOT, NUM, HD,IC)
      INTEGER ALPHA1, ASTAT, ASHOT
      REAL*8 RESULT(20), TT(4), X(4)
      DIMENSION ALPHA1(110,110), VPG(200), VAVPG(200), TTRMPG(200),
     1          KSITE(200), TUNC(2000), DUNC(2000),
     2          ASTAT(110), ASHOT(110), TSC(110,110), XSC(110,110),
     3          TCOR(110,110), XCOR(110,110), NS(110)
C
      LL = 0
      IST = 0
      DO 150 ICS = 1, NUM
        N = NS(ICS)
        IF (NUM .EQ. 71) WRITE (7,10) ASHOT(ICS), N
        IF (NUM .EQ. 110) WRITE (7,10) ASTAT(ICS), N
   10   FORMAT (1X, ///'PROCESSING ', A4, I5)
        IF (N .EQ. 0) GO TO 150

        L = 0
        DO 20 I = 1, N
          L = L + 1
          TUNC(L) = TSC(I,ICS)
   20   DUNC(L) = XSC(I,ICS)

C
C
C************* FIND APPARENT VELOCITY FOR THIS CMP'
C              IF N.LE.3 CANNOT USE MOVING AVERAGE
C              SO ASSIGN C THE GIVEN VELOCITY AND GOTO
C              DIFFERENT SECTION OF PROGRAMME
C
        IF (IC.EQ.0) GO TO 120
        IF (N .LE. 3) C = VADEF
        IF (N .LE. 3) GO TO 120


C
C     LOOP OVER ALL DATA AND CALCULATE VA FROM EACH 4 VALUES
C
        NSTRT = 1
        NEND = N - 3
        DO 110 L = NSTRT, NEND
          TT(1) = TUNC(L)
```

```
 30     TT(2) = TUNC(L + 1)
        TT(3) = TUNC(L + 2)
        TT(4) = TUNC(L + 3)
        X(1) = DUNC(L)
        X(2) = DUNC(L + 1)
        X(3) = DUNC(L + 2)
        X(4) = DUNC(L + 3)
        IFAIL = 0
        NN = 4
        CALL G02CAF(NN, X, TT, RESULT, IFAIL)
        IF (RESULT(6) .LE. 0.0) C = VADEF
        IF (RESULT(6) .LE. 0.0) GO TO 50
        C = 1 / RESULT(6)
        WRITE (7,40) C
 40     FORMAT (1X, 'RESULT FROM REGRESSION IS', F10.3)
        IF (N .EQ. 4) GO TO 120
 50     IF (C .LT. 4.0 .OR. C .GT. 12.0) C = VADEF
        WRITE (7,60) L, NSTRT, NEND, LL, C
 60     FORMAT (/'OUTPUT L, NSTRT, NEND,LL     '/, 4I5, /
   1            'APPARENT VELOCITY=', F5.2)


C
C     C  CALCULATED
C
        IF (L .EQ. NSTRT) LL = 0
 70     IF (L .GT. NSTRT) GO TO 80
        IF (LL .EQ. NSTRT) LL = L + 1
        IF (LL .EQ. 0) LL = L
        GO TO 90
 80     LL = LL + 1
C
C********** FIND EACH SHOT/STAT PAIR AND DO CORRECTION
C
 90     DO 100 J = 1, NUMC
100     IF (ALPHA1(LL,ICS) .EQ. KSITE(J)) IST = J
        IF (IST .EQ. 0) STOP
        VG = VPG(IST)
        V1 = VAVPG(IST)
        IF (C .LE. V1) C = VG + 0.1
        IF (VG .LE. V1) STOP
        IF ((C - VG) .LT. 0.1) C = VG + 0.1
        H1 = TTRMPG(IST) * V1 * VG / (VG**2 - V1**2) ** 0.5
        TAN1 = V1 / (C**2 - V1**2) ** 0.5
        TAN2 = VG / (C**2 - VG**2) ** 0.5
        COS2 = (1 - VG**2/C**2) ** 0.5
        COS1 = (1 - V1**2/C**2) ** 0.5
        TCOR(LL,ICS)=H1*(1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
        XCOR(LL,ICS) = H1 * (TAN1 - TAN2) + HD * TAN2
        WRITE (7,170) ALPHA1(LL,ICS), V1, VG, C, TTRMPG(IST), H1, HD,
   1    TCOR(LL,ICS), XCOR(LL,ICS)
        IF (L .EQ. NSTRT .AND. LL .EQ. NSTRT) GO TO 70
        IF (LL .EQ. NEND + 1) GO TO 70
        IF (LL  EQ. NEND + 2) GO TO 70
110   CONTINUE
```

```
        GO TO 150


C
C
C************ IF N.LE.3 READS C FROM FILE
C
C
C     LOOP OVER ALL DATA AND CALCULATE VA FROM EACH 4 VALUES
C
 120   IF (C .LT. 4.0 .OR. C .GT. 12.0) C = VADEF
       WRITE (7,160) C
       NSTRT = 1
       NEND = N
       DO 140 L = NSTRT, NEND
         LL = L
C
C********** FIND EACH SHOT/STAT PAIR AND DO CORRECTION
C
         DO 130 J = 1, NUMC
 130     IF (ALPHA1(LL,ICS) .EQ. KSITE(J)) IST = J
         IF (IST .EQ. 0) STOP
         VG = VPG(IST)
         V1 = VAVPG(IST)
         IF (C .LE. V1) C = VG + 0.1
         IF (VG .LE. V1) STOP
         IF ((C - VG) .LT. 0.1) C = VG + 0.1
         H1 = TTRMPG(IST) * V1 * VG / (VG**2 - V1**2) ** 0.5
         TAN1 = V1 / (C**2 - V1**2) ** 0.5
         TAN2 = VG / (C**2 - VG**2) ** 0.5
         COS2 = (1 - VG**2/C**2) ** 0.5
         COS1 = (1 - V1**2/C**2) ** 0.5
         TCOR(LL,ICS)=H1*(1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
         XCOR(LL,ICS) = H1 * (TAN1 - TAN2) + HD * TAN2
         WRITE (7,170) ALPHA1(LL,ICS), V1, VG, C, TTRMPG(IST), H1, HD,
   1     TCOR(LL,ICS), XCOR(LL,ICS)
 140   CONTINUE
 150 CONTINUE


 160 FORMAT (/'OUTPUT FROM SUBROUTINE SHALL'/'APPARENT VELOCITY=',
   1       F5.2)
 170 FORMAT (1X, A4, 1X, 6(F5.2,1X), ' CORT = ', F6.2, /, 47X,
   1       ' CORRSX = ', F6.2)
 180 FORMAT ('NEW DISTANCES FROM DMP AND XM25NU ARE ', 2F10.3)
 190 RETURN
     END
C
C*************************************************************
C
C                          MATCONPLT
```

```
C                          ---------
C
C*************************************************************
C
C
C     This programme will plot the cssp travel time and distance
C     matrices. The distance matrix can be plotted using either
C
      SUBROUTINE MATCOR(DST, DSH, TT, XX, INU)
      INTEGER*2 IUSER, FORMAT
      INTEGER ASHOT, ASTAT, XSTAT, XSHOT, AMAIN, KSTH
      DIMENSION TT(2000), ASHOT(110), ASTAT(110), IUSER(3), KSTH(180),
     1          DIST25(180), IST(110), FORMAT(5), ISH(110), XSTAT(110),
     2          XSHOT(110), AMAIN(60), ISTAT(60), TMAT(110,110),
     3          STDIST(110), SHDIST(110), X(110), XX(2000), DST(2000),
     4          DSH(2000), XDUM(110,110), TITLE(20), XDUMN(30,60),
     5          XDUMI(36,60), XDUMIN(71,60), XSH(71), XST(60)
      DATA NSEA /'N   '/
      DATA FORMAT /'''(', '3F', '10', '.4', ')'''/
      DATA IUSER /'''G', 'LL', '4'''/
      DATA AMAIN /'1   ', '2   ', '3   ', '4   ', '5   ', '6   ',
     1     '7   ', '8   ', '9   ', '10  ', '11  ', '12  ', '13  ',
     2     '14  ', '15  ', '16  ', '17  ', '18  ', '19  ', '20  ',
     3     '21  ', '22  ', '23  ', '24  ', '25  ', '26  ', '27  ',
     4     '28  ', '29  ', '30  ', '31  ', '32  ', '33  ', '34  ',
     5     '35  ', '36  ', '37  ', '38  ', '39  ', '40  ', '41  ',
     6     '42  ', '43  ', '44  ', '45  ', '46  ', '47  ', '48  ',
     7     '49  ', '50  ', '51  ', '52  ', '53  ', '54  ', '55  ',
     8     '56  ', '57  ', '58  ', '59  ', '60  '/
C
C
CCCCCCCCCCCCCCCCCCCC
C
C  INPUT  OF DATA
C
CCCCCCCCCCCCCCCCCCCC
C
   10 VAV = 0.0
      VREDP = 0.0
      WRITE (6,20)
   20 FORMAT (/'TITLE FOR OUTPUT'/'----------TITLE------....')
      READ (5,30) TITLE
   30 FORMAT (20A4)
      WRITE (7,30) TITLE
      WRITE (6,40)
   40 FORMAT (/'RED VEL OF PLOT (RETURN IF WANT TO PLOT TT/XX'/
     1        '_____.___')
      READ (5,50) VREDP
   50 FORMAT (F10.0)
      IF (VREDP .NE. 0.0) GO TO 80
      WRITE (6,60)
   60 FORMAT (/'INPUT AVERAGE CRUSTAL VELOCITY FOR T**2/X**2 PLOT')
      READ (5,70) VAV
   70 FORMAT (F10.4)
      VREDP = VAV
   80 WRITE (6,90)
   90 FORMAT (/'DIST SCALE CM/KM'/'_____.___')
      READ (5,100) DSCALE
  100 FORMAT (F10.0)
      DO 105 I=1,INU
        IF(VREDP.NE.0.0.AND.VAV.EQ.0.0) TT(I)= TT(I)- XX(I)/VREDP
        IF(VAV.NE.0.0) TT(I)= TT(I)**2 -(XX(I)/VAV) **2
  105   CONTINUE
C
CCCCCCCCCCCCCCCCCCCC
C
C
C     PLOTTING
C     --------
C
CCCCCCCCCCCCCCCCCCCC
C
      XMAXST = -999.0
      XMINST = 999.0
      XMAXSH = -999.0
      XMINSH = 999.0

      DO 110 I = 1, INU
        IF (DSH(I) .GT. XMAXSH) XMAXSH = DSH(I)
        IF (DSH(I) .LT. XMINSH) XMINSH = DSH(I)
        IF (DST(I) .GT. XMAXST) XMAXST = DST(I)
        IF (DST(I) .LT. XMINST) XMINST = DST(I)
  110 CONTINUE
      NUMPT = INU
      IF (INU .LT. 97) GO TO 130
      WRITE (6,120)
  120 FORMAT (/'WARNING TO MANY POINTS IN ARRAY TO BE CONTOURED<97')
C
C
C
C
C
C     SET UP SOURCE FILES FOR SURFACE.
C
C
C
C
  130 WRITE (6,140)
C
C
  140 FORMAT (//'-----------------------------------------------------------
     1-----'/, 10X, /'INPUT VALUES NEEDED BY SURFACE2'/
     2
     3'OPTIONS ARE PROVIDED FOR ALSO PLOTTING PERSPECTIVE DIAGRAMS'/
     4
     5'IF PLOT IS NOT QUITE CORRECT THE SURFACE2 COMMANDS CAN BE '/
     6        'ALTERED MANUALLY IN THE SOURCE FILE GENERATED BY THIS'/
     7        'PROGRAMME TO RUN WITH THE SURFACE2 PACKAGE'/
     8        ' OUTPUT OF SOURCE FILE IS IN UNIT 8'/
     9        ' OUTPUT OF DATA IN FORMAT FOR SURFACE2 IS IN UNIT 7'//
```

```
     *        '_____________________________________________________'
     1        )
C
C
      WRITE (1,150) IUSER
  150 FORMAT ('DEVICE        ', '1,', 3A2)
C
      WRITE (1,160) TITLE
  160 FORMAT ('TITLE          ', 20A4)
C
      WRITE (1,170) NUMPT, FORMAT
  170 FORMAT ('IDXY           ', I5, ',11,3,1,2,3,0,0,1,9999,', 5A2)
C
      WRITE (1,180) NUMPT, NUMPT
  180 FORMAT ('GRID           ', '0,', 2(I5,','))
C
      IMINSH = XMINSH
      IMINST = XMINST
      IMAXSH = XMAXSH + 1
      IMAXST = XMAXST + 1
      WRITE (1,190) IMINSH, IMAXSH, IMINST, IMAXST
  190 FORMAT ('EXTREMES      ', 3(I5,','), I5)
C
      WRITE (1,200)
  200 FORMAT ('CONTOUR       ')
C
      TTMIN = 999.0
      TTMAX = -999.0
      DO 210 I = 1, NUMPT
        IF (TT(I)  LT. TTMIN) TTMIN = TT(I)
        IF (TT(I) .GT. TTMAX) TTMAX = TT(I)
  210 CONTINUE
      TDIFF = TTMAX - TTMIN
      WRITE (6,220) TTMIN, TTMAX, TDIFF
  220 FORMAT (///, 'CONTOUR INTERVAL VALUES :TTMIN/TTMAX/TTDIFF =-',
     1        3F10.4, /,
     2        ' 1) INPUT BASE LEVEL OF CONTOURING (DEFAULT=TTMIN)'/,
     3        ' 2) INCREMENT STEP BETWEEN CONTOURS (CINCR)'/,
     4        ' 3) MAX NUMBER OF CONTOURS (FOR DEFAULT TYPE 0) (NOMAX)'/,
     5        ' 4) LABEL EVERY NTH CONTOUR - INPUT N '/,
     6        ' 5) HEIGHT OF NUMERALS (DEFAULT HN=0)'/,
     7        ' 6) NUMBER OF PTS AFTER DP (IDPT)'/,
     8
     9' 7) DIST BETWEEN LABELS ON SAME CONTOUR (DEFAULT DLABEL=0)'/,
     *        ' 8) PLOT EVERY NNTH CONTOUR AS HEAVY LINE '///,
     1        '   TTMIN    CINCR    NOMAX  N   HN  IDPT DLABEL NN'/,
     2        '_____.____.____.____.____.____.____.____.____.____')
C
      READ (5,230) TTMIN, CINCR, NOMAX, N, HN, IDPT, DLABEL, NN
  230 FORMAT (2F10.4, 2I5, F5.2, I2, F5.2, I2)
      WRITE (1,240) TTMIN, CINCR, NOMAX, N, HN, IDPT, DLABEL, NN
  240 FORMAT ('CINT          ', '0,', 2(F10.4,','), 2(I5,','), F5.2, ',',
     1        I2, ',', F5.2, ',', I2)
C
C     FOR A4 DSCALE=0.0 AND IS CALCULATED BY MAKING THE
C     SHOT AXES =6.0 INCHES.
C
      ICODE = 1
      WRITE (6,250) DSCALE
  250 FORMAT (/'DSCALE FOR SURFACE2 PLOT IS ', F10.4, ' CMS/KM')
      DSHSC = SHD / 2.54
      WRITE (1,260) ICODE, DSHSC
  260 FORMAT ('SIZCONTOUR    ', I1, ',', F5.2)
C
      WRITE (6,270) IMINSH, IMAXSH, IMINST, IMAXST
  270 FORMAT (/'AXES OF PLOT    :   SHOT MIN MAX, STAT MIN MAX-', 4I5, /,
     1        ' 1) DIST BETWEEN TICK MARKS ON SHOT AXES (SHT)'/,
     2        ' 2) FREQUENCY OF TICK MARKS ON SHOT AXES (SHF)'/,
     3        ' 3) DIST BETWEEN TICK MARKS ON STAT AXES (STT)'/,
     4        ' 4) FREQUENCY OF TICK MARKS ON STAT AXES (STF)'/,
     5        ' 5) HEIGHT OF NUMERALS (DEFAULT HN=0)'/,
     6        ' 6) NUMBER OF PTS AFTER DP (IDPT)'/,
     7
     8' 7) WHICH PLOT BORDERS TO LABEL 0=NONE 1=BOTTOM+LEFT 2=ALL '///,
     9        '  SHT  SHF  STT   STF  HN  IDPT N'/,
     *        '____.____.____.____.____.____.____.____.____.____')
C
      READ (5,280) SHT, ISHF, STT, ISTF, HN, IDPT, N
  280 FORMAT (2(F5.2,I2), F5.2, I5, I1)
      WRITE (1,290) SHT, ISHF, STT, ISTF, IDPT, N, HN
  290 FORMAT ('BOX           ', 2(F5.2,',',I2,','), I5, ',0,0,', I1, ',',
     1        F5.2)
C
      WRITE (1,330)
C
      WRITE (1,300)
  300 FORMAT ('OVERLAY       ')
C
      WRITE (1,190) IMINSH, IMAXSH, IMINST, IMAXST
      WRITE (1,290) SHT, ISHF, STT, ISTF, IDPT, N, HN
      WRITE (1,310) ICODE, DSHSC
  310 FORMAT ('BOXEXTREME    ', I1, ',', F5.2)
      WRITE (1,320)
  320 FORMAT ('POST          2,0,0,0.06,4')
      WRITE (1,330)
  330 FORMAT ('PERFORM       ')
      WRITE (1,340)
  340 FORMAT ('STOP          ')
      WRITE (6,350)
  350 FORMAT ('SOURCE FILE FOR SURFACE2 SET UP'/'COPYING DATA FOR ',
     1        'SURFACE2 INPUT TO UNIT 11')
      DO 360 I = 1, NUMPT
  360 WRITE (11,370) DSH(I), DST(I), TT(I)
  370 FORMAT (3F10.4)
      WRITE (11,380)
  380 FORMAT (/'DISTANCE DATA')
      DO 390 I = 1, NUMPT
  390 WRITE (11,370) DSH(I), DST(I), XX(I)
  400 STOP
      END
```

```
C
CA    AHJ Lewis
CD+
C*****************************************************************
C*                                                               *
C*        COMMON MID-POINT : TTERM PROGRAMME FOR REFLECTIONS     *
C*                                                               *
C*                                                               *
C* AHJ LEWIS      Dept. Geol. Sciences, Durham          Nov 1985 *
C*****************************************************************
C
C
C                    Description
C
C
C     This programme reads travel-times in the file assigned to unit5
C     which have been corrected for the shallow structure using the
C     programme TTCORR
C     The reflection equation T**2 = X**2/V**2 + 4 * H**2 / V**2
C     is fitted to the travel-times for a wide-angle reflected phase
C     using the method suggested in the paper by
C           Bamford DAL, 1978  Interpretatioan of Wide-angle reflections
C                              in Realistic Earth-Mantle Structures
C                              GJRAS
C
C                 ------------------------------
C
C
C     Programme reads distances of sites from M25B and sorts these
C     into CMP gathers and then fits each CMP tt gather to the reflection
C     equation. The average velocity and depth are then calculated and
C     outputted to unit 6. The results are then rearranged for plotting.
C
C     UNITS
C     -----
C
C     unit 5  - the corrected travel-times - EXAMPLE INPUT in TTRM-CSP.PMP
C
C               1   TITLE                        (A4)
C               2   INU, HD, NSTCMP, NUM         (I5,F5.2,2I5)
C                   INU    - number of travel times
C                   HD     - depth of datum travel-times corrected to
C                   NSTCMP - start cmp distance from M25B
C                   NUM    - number of CMP's to do
C
C
C               3   DST,DSH,TT,XX,ASTAT,ASHOT  (4F10.4,1X,2A4)
C                   repeated INU times
C
C     unit 6  - output of results
C     unit 7  - output of values read from unit5
C     unit 9  - plotfile of results
CD-
      REAL*8 A, B, SUMTT, SUMCD, C, D, TOTCD, TOTDD, V, TOTRS2, SETTRM
      DIMENSION NSTAT(400), NSHOT(400), ALPHA1(2000), ALPHA2(2000),
```

```
     1          T(2000), DIST(2000), TERR(2000), DISTER(2000),
     2          KSTH(180), DIST25(180), NCMP(2000), XCMP(400),
     3          TCMP(2000), DCMP(2000), DMP(2000), SUMTT(400),
     4          SUMDD(400), TT(2000), XX(2000),DST(2000),DSH(2000),
     5          RESID(2000), D(2000), ASTAT(2000), ASHOT(2000),A(400),
     6          B(400), TTRM(400), C(2000), XANOT(20), ANOT(20),
     7          TITLE(20), H(400), TITLES(20), TTRMP(400), HP(400),
     8          XP(400), SETTRM(400), HERR(400), KSITE(400), VG(400),
     9          VAVPG(400), TTRMPG(400), TTEST(20), TRESID(2000)
      INTEGER ASTAT, ASHOT, BLANK, ALPHA1, ALPHA2, ANOT
      DATA M25B /'M25B'/
      DATA TTEST /12.7, 4.3, 3.18, 2.78, 5.57, 2.45, 2.37, 2.31, 2.26,
     1     2.23, 2.2, 2.18, 2.16, 2.15, 2.13, 2.12, 2.11, 2.10, 2.09,
     2     2.09/


C
C
C************************** READ IN DATA FROM ************************
C
C
      READ (5,450) TITLE
      WRITE (6,450) TITLE
      WRITE (7,450) TITLE
      READ(5,430) INU,HD,NSTCMP,NUM
      WRITE(7,430) INU,HD,NSTCMP,NUM
      DO 10 I=1,INU
        READ(5,431) DST(I),DSH(I),TT(I),XX(I),ASTAT(I),ASHOT(I)
   10   WRITE(7,431) DST(I),DSH(I),TT(I),XX(I),ASTAT(I),ASHOT(I)
  431 FORMAT(4F10.4,1X,2A4)
   30 CALL FTNCMD('ASSIGN 10=CPT9:XMAT-M25B', 24)
      READ (10,450) TITLES
      READ (10,460) (KSTH(I),I=1,180)
      READ (10,470) (DIST25(I),I=1,180)


C
C
C                        **************
C********************                  ****************************
C******************** LOOP OVER CMPS ****************************
C********************                  ****************************
C                        **************
C
C
   40 NTOT = 0
      TOTDD = 0.0
      TOTCD = 0.0
      TOTRS2 = 0.0
      DO 160 ICMP = 1, NUM
C
C
C *************** FIND NEW XCMP
C
C
        SUMTT(ICMP)=0.0
        SUMDD(ICMP) = 0.0
        XCMP(ICMP) = NSTCMP+ICMP-1.0
        WRITE(6,490) XCMP(ICMP)
        WRITE(7,490) XCMP(ICMP)


C
C
C************** FIND STATION SHOT PAIRS WITH XCMP(ICMP)
C
C     STATION SPACING IS 2KM
C     CMP SPACING IS 1KM <=> +/- 0.5KM !
C
   60   XU = XCMP(ICMP) + 0.5
        XD = XCMP(ICMP) - 0.5


C
C     LOOP OVER DATA TO FIND XCMPS
C
        NCMP(ICMP) = 0
        DO 100 I = 1, INU
            XMP = (DST(I) + DSH(I)) / 2.0


C
C     IF XMP NOT EQUAL TO NEW XCMP TRY NEXT STAT/SHOT PAIR
C
            IF (XMP .EQ. 0.0) GO TO 100
            IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 100


C                 ____________________________
C ************* FOUND STAT/SHOT PAIR WITH XMP=XCMP ******************
C                 ____________________________
C
```

```
 80         NOMP(IOMP) = NOMP(IOMP) + 1
            NTOT = NTOT + 1
            ALPHA1(NTOT) = ASTAT(I)
            ALPHA2(NTOT) = ASHOT(I)
            TOMP(NTOT) = TT(I)
            DOMP(NTOT) = XX(I)
            DMP(NTOT) = XMP
            SUMTT(IOMP) = SUMTT(IOMP) + TOMP(NTOT) ** 2
            SUMDD(IOMP) = SUMDD(IOMP) + DOMP(NTOT) ** 2
            WRITE (7,510) ASTAT(IOMP),ASHOT(IOMP),DOMP(NTOT), TOMP(NTOT)
            GO TO 100
 100    CONTINUE


C
C
C                 ******************************
C****************                                  *******************
C**************** CALCULATE LEAST SQUARES TOTALS *******************
C****************                                  *******************
C****************      IF NOMP(IOMP)> OR = 2       *******************
C****************      IF NOT RESET VARIABLES      *******************
C****************                                  *******************
C                 ******************************
C
C
 130    WRITE (6,380) NOMP(IOMP), NTOT
        IF (NOMP(IOMP) .GE. 2) GO TO 140
        IF (NOMP(IOMP) .EQ. 0) GO TO 160
        TOMP(NTOT) = 0.0
        DOMP(NTOT) = 0.0
        NTOT = NTOT - 1
        GO TO 160
C
 140    B(IOMP) = SUMDD(IOMP) / NOMP(IOMP)
        A(IOMP) = SUMTT(IOMP) / NOMP(IOMP)
        N = NOMP(IOMP)
        ITOT = NTOT - N
        DO 150 IJ = 1, N
          ITOT = ITOT + 1
          D(ITOT) = DOMP(ITOT) ** 2 - B(IOMP)
          C(ITOT) = TOMP(ITOT) ** 2 - A(IOMP)
          TOTDD = TOTDD + D(ITOT) ** 2
 150    TOTCD = TOTCD + C(ITOT) * D(ITOT)
C        WRITE (6,270) TOTCD, TOTDD
C
C
        IF (ITOT .NE. NTOT) STOP
C
C
 160 CONTINUE
```

```
C
C                 ***********************************
C****************                                  *******************
C**************** CALCULATE LEAST SQUARES VELOCITY *******************
C****************                                  *******************
C                 ***********************************
C
C
      IF (TOTDD .GT. 0.0 .OR. TOTCD .GT. 0.0) GO TO 170
      WRITE (6,400)
      STOP
 170  VV = (TOTDD/TOTCD)
      V = VV ** 0.5
      WRITE (6,410) V


C
C
C********** CALCULATE ERRORS OVER THE XOMPS USED *******************
C
C
      L = 0
      ITOT = 0
      DO 190 I = 1, NUM
        IF (NOMP(I) .LT. 2) GO TO 190
        L = L + 1
        N = NOMP(I)
        SETTRM(I) = 0.0
        DO 180 J = 1, N
          ITOT = ITOT + 1
          RESID(ITOT) = C(ITOT) - D(ITOT) / VV
          SETTRM(I) = SETTRM(I) + RESID(ITOT) ** 2
          TOTRS2 = TOTRS2 + RESID(ITOT) ** 2
 180    CONTINUE
        TTRM(I) = A(I) - B(I) / VV
        DF = NOMP(I) * (NOMP(I) - 1)
        SETTRM(I) = SETTRM(I) / DF
        SETTRM(I) = (SETTRM(I)) ** 0.5
 190  CONTINUE
      NUMP = L
      DF = NTOT - NUMP - 1
      SE = TOTRS2 / DF
      SE = SQRT(SE)
      SEV = SE * VV / TOTDD ** 0.5
      WRITE (6,420) V, SEV, SE
```

```
C
C
C********** CALCULATE TTRM AND DEPTHS FOR XOMPS ********************
C
C
      WRITE (6,290)
      L = 0
      DO 220 I = 1, NUM
        IF (NOMP(I) .LT  2) GO TO 210
        IF (TTRM(I) .LT. 0.0) GO TO 200
        L = L + 1
        H(I) = TTRM(I) * VV / 4.0
        TTRM(I) = TTRM(I) ** 0.5
        H(I) = H(I) ** 0.5
        IF (HD.NE.0.0) H(I) = H(I) + HD
        HERR(I) = SETTRM(I) ** 2 + SEV ** 2
        SETTRM(I) = SETTRM(I) / 2.0
        HERR(I) = HERR(I) / 2.0
        IWORK = NOMP(I)
        IF (IWORK .GT. 20) IWORK = 20
        HERR95 = HERR(I) * TTEST(IWORK)
        TTRMP(L) = TTRM(I)
        HP(L) = H(I)
        XP(L) = XOMP(I)
        WRITE (6,300) XOMP(I), TTRM(I), SETTRM(I),
     1  H(I), HERR(I), HERR95
        GO TO 220
 200    TTRM(I) = NOMP(I)
        H(I) = NOMP(I)
        WRITE (6,320) XOMP(I)
 210    TTRM(I) = NOMP(I)
        H(I) = NOMP(I)
        WRITE (6,310) XOMP(I)
 220 CONTINUE
      NUMP = L


C
C
C*********** WRITE OUT RESIDUALS ON EACH TRAVEL TIME USED ***********
C
C
      WRITE (6,330)
      I = 0
      DO 240 IOMP = 1, NUM
        IF (NOMP(IOMP) .LT. 2) GO TO 240
        N = NOMP(IOMP)
        WRITE (6,490) XOMP(IOMP)
        DO 230 K = 1, N
          I = I + 1
          TSOL = TTRM(IOMP) ** 2 + DOMP(I) ** 2 / VV
          TSOL = TSOL ** 0.5
          TRESID(I) = TOMP(I) - TSOL
          WRITE (6,340) ALPHA1(I), ALPHA2(I), DOMP(I), TOMP(I),
     1    RESID(I), TRESID(I)
 230    CONTINUE
 240 CONTINUE


C
C
C          ******* PLOT RESIDUALS ******
C
      CALL PAPER(1)
      YN = 0.0
      NDIM = 2000
      NOCHAR = 45
      CALL PLT(DOMP, RESID, YN, DS, TS, NTOT, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'DISTANCE (KM)', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(6.0, 8.2, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.2, HD, 1)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, V, 2)
      CALL PLOTCS(6.0, 0.2, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.2, SE, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'RESIDUALS', 9)
      CALL CTRORI(0.0)
      CALL FRAME
      YN = 0.0
      NDIM = 2000
      NOCHAR = 45
      CALL PLT(DOMP, TRESID, YN, DS, TS, NTOT, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'DISTANCE (KM)', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(6.0, 8.2, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.2, HD, 1)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, V, 2)
      CALL PLOTCS(6.0, 0.2, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.2, SE, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'CORR TT  RESIDUALS',
     1     18)
      CALL CTRORI(0.0)


C
```

```
C           ******* PLOT ANNOTATION ******
C
      READ (5,350) NANOT
      READ (5,360) (ANOT(II),II=1,NANOT)
C
      DO 250 II = 1, NANOT
        DO 250 JJ = 1, 180
          IF (ANOT(II) .EQ. KSTH(JJ)) XANOT(II) = DIST25(JJ)
  250 CONTINUE
C
C           ******* PLOT TTERMS ******
C
  260 YN = -1.0
      NDIM = 200
      AMN = SEV
      NOCHAR = 43
      CALL FRAME
      CALL PLT(XP, TTRMP, YN, DS, TS, NUMP, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 270 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
  270 CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B OF OMP)', 26)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(6.0, 8.5, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.5, HD, 1)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, V, 2)
      CALL PLOTCS(6.0, 0.32, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.32, SE, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMN, 4)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'CORR REFLECT. TIME-TERM
     1S-DATUM', 30)
      CALL CTRORI(0.0)
C
C           ******* PLOT DEPTHS ******
C
      YN = -2.0
      NDIM = 200
      NOCHAR = 43
      CALL FRAME
      CALL PLT(XP, HP, YN, DS, TS, NUMP, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 280 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
  280 CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B OF OMP)', 26)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(6.0, 8.5, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.5, HD, 1)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, V, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMN, 4)
      CALL PLOTCS(6.0, 0.32, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.32, SE, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, '  CORR DEPTH (KM)', 17)
      CALL CTRORI(0.0)
C
C
C********************** FORMAT STATEMENTS **********************
C
C
C
  290 FORMAT (/////'TTERMS FOR THE OMPS USED'/'_____________________
     1____'///
     2
     3'STAT SHOT   XOMP(M25B)  TTRM     ERROR     DEPTH+ERROR       95%
     4ERROR '/)
  300 FORMAT (1X, 6F10.3)
  310 FORMAT (1X, F10.3, ' THIS OMP NOT USED')
  320 FORMAT (1X, F10.3, ' THIS OMP TTERM<0!*************')
  330 FORMAT (////'************* RESIDUALS ***************'//
     1        ' STAT  SHOT    DIST      TT          RESID '/)
  340 FORMAT (1X, A4, 1X, A4, 4F10.3)
  350 FORMAT (I2)
  360 FORMAT (20A4)
  370 FORMAT (1X, A4, '/', A4, F10.3)
  380 FORMAT (/' NUMBER= ', 2I5)
  390 FORMAT (/' TOTCD,TOTDD= ', 2F15.3)
```

```
 400 FORMAT (/'UNABLE TO CALCULATE LEAST SQUARES VELOCITY')
 410 FORMAT (/'LEAST SQUARE VELOCITY IS ', F10.3)
 420 FORMAT (//////' LEAST SQUARE VELOCITY = ', F5.3, '+/-', F7.4, /
    1        'STANDARD ERROR OF SOLUTION IS ', F10.4)
 430 FORMAT (I5,F5.2,2I5)
 440 FORMAT ((20A4))
 450 FORMAT (20A4)
 460 FORMAT (3(60A4))
 470 FORMAT (3(60F10.4))
 480 FORMAT ((A4,1X,3F10.3))
 490 FORMAT (///'COMMON MID-POINT ', F10.4,/
    1        '_____________________________________')
 500 FORMAT (/'DID NOT FIND CMP')
 510 FORMAT (1X, A4, '/', A4, 2F10.3)


C                                ____
C                      ********* END *********
C                                ____
 520 CALL GREND
     STOP
     END



C
C*****************************************************************
C
C
C ********************** PLT ***********************************
C
C
C*****************************************************************
C
C
     SUBROUTINE PLT(X, Y, YN, DS, TS, N, NDIM, NOCHAR)
C
C
     DIMENSION X(NDIM), Y(NDIM)
     XMAX = -9999.9
     XMIN = 9999.9
     YMAX = -9999.9
     YMIN = 9999.9
     DO 10 I = 1, N
       IF (X(I) .GT. XMAX) XMAX = X(I)
       IF (X(I) .LT. XMIN) XMIN = X(I)
       IF (Y(I) .GT. YMAX) YMAX = Y(I)
  10 IF (Y(I) .LT. YMIN) YMIN = Y(I)
     IF (YN .LT. 0.0) YMIN = 0.0
     IF (YN .EQ. - 2.0) XMAX = 400.0
     IF (YN .EQ. - 2.0.AND. XMIN.GE.0.0) XMIN =  0.0
     XMAX = XMAX + 0.1 * (XMAX - XMIN)
     XMIN = XMIN - 0.1 * (XMAX - XMIN)
     YMAX = YMAX + 0.1 * (YMAX - YMIN)
     YMIN = YMIN - 0.1 * (YMAX - YMIN)
     IF (YN .LT. 0.0) YMIN = 0.0
     IF (YN .EQ. - 2.0) YMAX = 50.0
     DS = 9.5 * 2.54
     TS = 6.0 * 2.54
     CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     IF (YN .GE. 0) CALL MAP(XMIN, XMAX, YMIN, YMAX)
     IF (YN .LT. 0) CALL MAP(XMIN, XMAX, YMAX, YMIN)
     CALL CTRMAG(15)
     CALL SCALES
     CALL BORDER
     CALL PTPLOT(X, Y, 1, N, NOCHAR)
     CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     CALL MAP(XMIN, XMAX, 0.0, 10.0)
     RETURN
     END
C
C
```

```
C
CA    AHJ Lewis
CD+
C*****************************************************************
C*                                                               *
C*                  TTERM PROGRAMME FOR REFRACTIONS              *
C*                                                               *
C*                                                               *
C* AHJ LEWIS      Dept. Geol. Sciences, Durham          Nov 1985 *
C*****************************************************************
C
C
C                      Description
C
C
C     This programme reads travel-times a travel time matrix and fits these
C     to the refraction equation using the method of least squares.
C     INVERSION OF SYMMETRIC MATRIX USING A CHOLESKY ALGORITHM
C     IS CARRIED OUT BY CALLING NAG ROUTINE F01ABF
C
C     The original programme was written by Swinburne (1975) and modified by
C     Summers (1982). The version now available gives the ability of
C     reading the travel-times from the travel time matrices. The input is kept
C     together in one block. Alteration for other data is thus simplified.
C     Also the results are now plotted using ghost routines.
C
C  INPUT ON UNIT 5  - example provided in the file TTRM-CSSP.FN
C
C
C
C  OUTPUT ON UNIT 6
C
C  CONSISTS OF  TIME TERM SOLUTION FOR UPTO 4 CONSTRAINED VELOCITIES
C  AND THE LEAST SQUARES VELOCITY.  LIST OF TIME TERMS ARE GIVEN AND
C  THE ORIGINAL CARDS WITH RESIDUALS  ETC. AND THE VARIANCE-COVARIANC
C  MATRIX TIMES  100.
C
C
C  OUTPUT ON UNIT 7
C
C    N,M,K,((VEL(I),VELERR(I)),I=1,4)...(3(5X,I5),8F5.2)
C                        N= NUM OF TIME TERMS
C                        M= NUM OF TRAVEL TIMES
C                        K= NUM OF CONSTRAINT TIME TERMS
C                  VEL(I)= UPTO 4 CONSTRAINT VELOCITIES
C               VELERR(I)= ERROR IN CONSTRAINT VELOCITIES
C
C    (ASITE(I),I=1,N) SITE NAMES (FORMAT 4X,A4)
C    ONE LINE PER SITE NAME USED IN NUMBERING TIME-TERMS
C    ENTER NAMES IN ORDER TO BE NUMBERED
C
C    K CONSTAINT CARDS  ALPHA(I),TTRM(I),TRERR(I)
C                  TTRM(I)= TIME TERM OF STATION
C                 TRERR(I)=  ERROR IN TIME TERM
C                 ALPHA(I)= NAME OF STATION
C
C    M TRAVEL TIME DATA  ALPHA1(I),ALPHA2(I),TT(I),TTERR(I),
C                        DIST(I),DISTER(I)
C                 ALPHA1(I)= NAME OF SHOT STATION
C                 ALPHA2(I)= NAME OF RECORDING STATION
C                     TT(I)= TRAVEL TIME
C                  TTERR(I)= ERROR IN TRAVEL TIME
C                   DIST(I)= DISTANCE
C                DISTERR(I)=  ERROR IN THE DISTANCE
C
C    THIS OUTPUT CAN BE USED AS INPUT TO ANY OF THE PREVIOUS
C    TIME-TERM PROGRAMMES WRITTEN BY SWINBURNE OR SUMMERS.
C
C     MATRIX E= PART OF THE TIME TERM NOT DEPENDENT ON VELOCITY
C     MATRIX F= PART OF TIME TERM (DISTANCE) VELOCITY DEPENDENT
C B,W ST.ERROR= STANDARD ERROR ON THE TIME TERM USING THE BERRY AND
C               WEST  FORMULAE
C      95 CONF= 95% CONFIDENCE LIMIT ON THE TIME TERM ESTIMATED FROM
C               THE MATRIX (ATA)-1
C     NUM DATA= NUMBER OF DATA TO  A STATION
C      MATIX C= PART OF TT NOT DEPENDENT OF VELOCITY
C     MATRIX D= PART OF TT DISTANCE VELOCITY DEPENDENT.  IE REVERSED
C
C
      IMPLICIT INTEGER(I - N)
      INTEGER Q
      REAL*4 KSTH
      DOUBLE PRECISION ATA, B, Z(200), ATEMP
      DIMENSION TTRM(25), TRERR(25), LA(2025), E(200), F(200),
     1          EERR(200), FERR(200), C(2000), D(2000), DERR(2000),
     2          TTSE(3200), CERR(2000), NTTSE(200), RESID(2000),
     3          RESERR(2000), TMTRM(200), TMTERR(200), EERG(200),
     4          FERG(200), CERG(2000), DERG(2000), TMTERG(200),
     5          RESERG(2000), X25(200), KSTH(170), DIST25(170),
     6          ANOT(99), XANOT(99), TITLE(20), H(200), VPG(200),
     7          VAVPG(200), TTRMPG(200), Q(200,200)
C
      DIMENSION TTEST(20)
      DIMENSION VEL(25), VELERR(25), VELERG(25)
      COMMON /ATAB/ ATA(200,200), B(200,200)
      COMMON /DATA/ TT(2000), TTERR(2000), DIST(2000), DISTER(2000)
      COMMON /NAME/ ALPHA(25), ALPHA1(2000), ALPHA2(2000), ASITE(200)
      COMMON /NUM/ NUM1(2000), NUM2(2000), NUM(25)
      DATA TTEST /12.7, 4.3, 3.18, 2.78, 2.57, 2.45, 2.37, 2.31, 2.26,
     1     2.23, 2.2, 2.18, 2.16, 2.15, 2.13, 2.12, 2.11, 2.10, 2.09,
     2     2.09/
C
C
C
C*********** READ IN M  EQUATIONS - UPTO 2000  ALLOWED ***************
C*********** READ IN N  STATIONS  - UPTO  200  ALLOWED ***************
C
C
C
      READ (5,790) TITLE
```

```
      WRITE (6,790) TITLE
      WRITE (7,790) TITLE
      CALL MATRD(ALPHA1, ALPHA2, ASITE, TT, TTERR, DIST, DISTER, M, N,
     1     VPG, VAVPG, TTRMPG)
      DO 10 I = 1, 25
   10 NUM(I) = 0
      READ (5,800) K, (VEL(I),VELERR(I),I=1,4)
      WRITE (6,800) K, (VEL(I),VELERR(I),I=1,4)
      DO 20 IJK = 1, 4
   20 VELERG(IJK) = VELERR(IJK)
      IF (K) 50, 50, 30
   30 DO 40 I = 1, K
        READ (5,670) ALPHA(I), TTRM(I), TRERR(I)
   40 CONTINUE
   50 CONTINUE
C
C
C
C********************  ZERO ARRAYS  ******************************
C
C
C
      DO 60 I = 1, N
        DO 60 J = 1, N
          B(I,J) = 0.0
          ATA(I,J) = 0.0
   60 CONTINUE
C
C
C**** NUMBER THE INPUT EQUATIONS WITH LOCATIONS IN ASITE *****
C
C
      DO 80 I = 1, M
        NUM1(I) = 0
        NUM2(I) = 0
        DO 70 J = 1, N
          IF (ALPHA1(I) .EQ. ASITE(J)) NUM1(I) = J
          IF (ALPHA2(I) .EQ. ASITE(J)) NUM2(I) = J
   70   CONTINUE
   80 CONTINUE


C
C
C**********************  FORM ATA  ******************************
C
C                DIAGONALS = NUM OF TIMES SITE IS USED
C                ATA(I,J)  = NUM OF TIMES ASITE(I) IS
C                            LINKED WITH ASITE(J).
C                 ATA(I,J) = ATA(J,I)
C
C
      IF (K) 130, 130, 90
   90 DO 120 I = 1, K
        DO 100 J = 1, N
  100   IF (ALPHA(I) .EQ. ASITE(J)) NUM(I) = J
        K1 = NUM(I)
        IF (K1 .NE. 0) GO TO 110
        WRITE (6,720) ALPHA(I)
        STOP
  110   ATA(K1,K1) = ATA(K1,K1) + 1.0
  120 CONTINUE
  130 CONTINUE
      DO 160 I = 1, M
        K1 = NUM1(I)
        K2 = NUM2(I)
        IF (K1 .EQ. 0 .OR. K2 .EQ. 0) GO TO 140
        IF (K1 .NE. K2) GO TO 150
        WRITE (6,730) I, K1, K2
        STOP
  140   WRITE (6,740) I, ALPHA2(I), ALPHA1(I), K1, K2
        STOP
  150   ATA(K1,K2) = ATA(K1,K2) + 1.0
        ATA(K2,K1) = ATA(K1,K2)
        ATA(K1,K1) = ATA(K1,K1) + 1.0
        ATA(K2,K2) = ATA(K2,K2) + 1.0
  160 CONTINUE


C
C
C*******************  OUTPUT DATA TO UNIT 7  ************************
C
C                  THIS OUTPUT CAN BE USED AS INPUT
C                  TO THE OTHER TTERM PROGRAMMES.
C
C
C
      WRITE (7,750) N, M, K, (VEL(I),VELERR(I),I=1,4)
      WRITE (7,660) (ASITE(J),J=1,N)
      IF (K) 190, 190, 170
  170 DO 180 I = 1, K
        WRITE (7,670) ALPHA(I), TTRM(I), TRERR(I), ASITE(NUM(I)), NUM(I)
  180 CONTINUE
  190 DO 200 I = 1, M
        WRITE (7,680) ALPHA1(I), ALPHA2(I), TT(I), TTERR(I), DIST(I),
     1  DISTER(I), ASITE(NUM1(I)), ASITE(NUM2(I)), NUM1(I), NUM2(I)
  200 CONTINUE
C
C     OUTPUT  NORMAL MATRIX
C
      WRITE (6,690)
      DO 210 I = 1, N
        DO 210 J = 1, N
          Q(I,J) = ATA(I,J)
          IF (I .EQ. J .AND. Q(I,J) .EQ. 0) WRITE (6,700) ASITE(I)
  210 CONTINUE
  220 N1 = 50
      IF (N .LT. 50) N1 = N
      NS2 = 51
```

```
      N2 = 100
      IF (N .LT. 100) N2 = N
      NS3 = 101
      N3 = 150
      IF (N .LT. 150) N3 = N
      WRITE (6,230)
  230 FORMAT (///'NORMAL MATRIX')
      WRITE (7,850)
      DO 240 I = 1, N
  240 WRITE (6,840) ASITE(I), (Q(I,L),L=1,N1)
      IF (N .LT. NS2) GO TO 270
      WRITE (6,860)
      DO 250 I = 1, N
        WRITE (6,840) ASITE(I), (Q(I,L),L=NS2,N2)
  250 CONTINUE
      IF (N .LT. NS3) GO TO 270
      WRITE (6,870)
      DO 260 I = 1, N
        WRITE (6,840) ASITE(I), (Q(I,L),L=NS3,N3)
  260 CONTINUE
  270 CONTINUE



C
C
C
C******************** INVERT ATA IF REQUIRED ************************
C
C
C
C
      WRITE (6,710)
      READ (5,620) IQ
      IF (IQ .EQ. 0) STOP
      IFAIL = 0
      IA = 200
      IB = 200
C
C
C      CALL MATRIX INVERSION ROUTINE F01ABF FROM *NAG
C      ____________________________________________
C
      CALL F01ABF(ATA, IA, N, B, IB, Z, IFAIL)
C
C      ONLY BOTTOM HALF OF B(I,J) IE  J<I HAS
C      INVERTED MATRIX VALUES SO FILL UP
C      J>I LOCATIONS WITH I>J VALUES
C
      DO 280 I = 1, N
        DO 280 J = 1, I
  280 B(J,I) = B(I,J)
```

```
C
C
C
C************* CALCULATION  OF MATRICES E AND F ********************
C
C
C                E=(ATA-1)(A)T(TT)        TT = TRAVEL TIME
C                F=(ATA-1)(A)T(DIST)   DIST = DISTANCE
C                EERR, FERR ARE ERRORS
C
C
      DO 330 I = 1, N
        E(I) = 0.0
        F(I) = 0.0
        EERR(I) = 0.0
        FERR(I) = 0.0
        EERG(I) = 0.0
        FERG(I) = 0.0
C
C        SUM CONSTRAINTS
C
        IF (K) 310, 310, 290
  290   DO 300 J = 1, K
          K1 = NUM(J)
          E(I) = E(I) + B(I,K1) * TTRM(J)
          AMN = B(I,K1) * TRERR(I)
          EERG(IN) = EERG(IN) + ABS(AMN) * ABS(AMN)
          EERR(IN) = EERR(IN) + ABS(AMN)
  300   CONTINUE
  310   CONTINUE
C
C        SUM M EQUATIONS
C
        DO 320 J = 1, M
          K1 = NUM1(J)
          K2 = NUM2(J)
          R = B(I,K1) + B(I,K2)
          E(I) = E(I) + R * TT(J)
          F(I) = F(I) + R * DIST(J)
          AMN = R * TTERR(J)
          EERR(I) = EERR(I) + ABS(AMN)
          EERG(I) = EERG(I) + ABS(AMN) * ABS(AMN)
          AMN = R * DISTER(J)
          FERG(I) = FERG(I) + ABS(AMN) * ABS(AMN)
  320   FERR(I) = FERR(I) + ABS(AMN)
        EERG(I) = SQRT(EERG(I))
  330 FERG(I) = SQRT(FERG(I))



C
C
C*************** CALCULATION OF MATRICES C AND D ****************
C
```

```
C
C
C
C
      TOTCD = 0.0
      TOTDD = 0.0
      TCDERR = 0.0
      TDDERR = 0.0
      TCDERG = 0.0
      TDDERG = 0.0
      DO 340 I = 1, M
        C(I) = TT(I) - E(NUM1(I)) - E(NUM2(I))
        D(I) = DIST(I) - F(NUM1(I)) - F(NUM2(I))
        TOTCD = TOTCD + C(I) * D(I)
        TOTDD = TOTDD + D(I) * D(I)
        CERR(I) = TTERR(I) + EERR(NUM1(I)) + EERR(NUM2(I))
        AMN = TTERR(I) ** 2 + EERG(NUM1(I)) ** 2 + EERG(NUM2(I)) ** 2
        CERG(I) = SQRT(AMN)
        DERR(I) = DISTER(I) + FERR(NUM1(I)) + FERR(NUM2(I))
        AMN = DISTER(I) ** 2 + EERG(NUM1(I)) ** 2 + EERG(NUM2(I)) ** 2
        DERG(I) = SQRT(AMN)
        TCDERR = TCDERR + C(I) * DERR(I) + D(I) * CERR(I)
        TCDERG = TCDERG + (C(I)*DERG(I)) ** 2 + (D(I)*CERG(I)) ** 2
        TDDERR = TDDERR + 2.0 * (D(I)*DERR(I))
  340 TDDERG = TDDERG + 2.0 * (D(I)*DERG(I)) ** 2
      TCDERG = SQRT(TCDERG)
      TDDERG = SQRT(TDDERG)


C
C
C************** CALCULATION OF LEAST SQUARES VELOCITY **************
C
C
C
C
      IF (TOTCD) 350, 350, 360
  350 WRITE (6,760)
      VEL(5) = 0.0
      GO TO 370
  360 VEL(5) = TOTDD / TOTCD
      VELERR(5) = TDDERR / TOTCD + (TOTDD*TCDERR) / (TOTCD*TOTCD)
      AMN = (TDDERG/TOTCD) ** 2 + ((TOTDD*TCDERG)/(TOTCD*TOTCD)) ** 2
      VELERG(5) = SQRT(AMN)
      WRITE (6,770) VEL(5), VELERR(5), VELERG(5)
      WRITE (6,780) TOTDD, TDDERR, TDDERG
  370 CONTINUE


C
C
C******************** FOR EACH VEL(J) ************************
C**********  CALCULATE RESIDUALS, STANDARD ERRORS **************
```

```
C******************** PLOT RESULTS ************************
C
C
C
C
C
      MEM = 0
      DO 610 J = 1, 5
        IF (VEL(J)) 380, 610, 380
  380   DO 390 I = 1, N
          TTSE(I) = 0.0
  390   NTTSE(I) = 0
        TOTRS2 = 0.0
        DO 400 I = 1, M
          RESID(I) = C(I) - D(I) / VEL(J)
          RESERR(I) = CERR(I) + DERR(I) / VEL(J) + (D(I)*VELERR(J)) / (
     1    VEL(J)**2)
          AMN = CERG(I) ** 2 + (DERG(I)/VEL(J)) ** 2 + ((D(I)*VELERG(J))
     1    /(VEL(J)**2)) ** 2
          RESERG(I) = SQRT(AMN)
          TOTRS2 = TOTRS2 + RESID(I) ** 2
          NTTSE(NUM1(I)) = NTTSE(NUM1(I)) + 1
          NTTSE(NUM2(I)) = NTTSE(NUM2(I)) + 1
          TTSE(NUM1(I)) = TTSE(NUM1(I)) + RESID(I) ** 2
  400   TTSE(NUM2(I)) = TTSE(NUM2(I)) + RESID(I) ** 2
        DO 410 I = 1, N
          AMN = 0.0
          IF (NTTSE(I) .LE. 1) GO TO 410
          AMN = TTSE(I) / (NTTSE(I)*(NTTSE(I) - 1))
  410   TTSE(I) = SQRT(AMN)
C
C
C       IF VEL(J) IS LEAST SQUARES VELOCITY DEGREES OF FREEDOM=M-N+K-1
C       IF J<5 THEN VELOCITY IS GIVEN SO DEGREES OF FREEDOM = M-N+K
C
C
        IF (J - 5) 430, 420, 430
  420   MN = M - N + K - 1
        GO TO 440
  430   MN = M - N + K
  440   AMN = TOTRS2 / MN
        SIGSOL = SQRT(AMN)
        DO 460 I = 1, N
          DO 450 JJ = 1, N
  450     B(I,JJ) = AMN * B(I,JJ)
  460   CONTINUE
        DO 470 I = 1, N
          AMN = (EERG(I)) ** 2 + (FERG(I)/VEL(J)) ** 2 + ((VELERG(J)*F(
     1    I))/(VEL(J)*VEL(J))) ** 2
          TMTERG(I) = SQRT(AMN)
          TMTRM(I) = E(I) - F(I) / VEL(J)
  470   TMTERR(I) = EERR(I) + FERR(I) / VEL(J) + (VELERR(J)*F(I)) / (
     1  VEL(J)*VEL(J))
        IF (J - 5) 490, 480, 490
  480   AMN = (VEL(5)**2*SIGSOL) / SQRT(TOTDD)
```

```
      AMNV = (VEL(5)**2*SIGSOL) / SQRT(TOTDD)
      WRITE (6,810) AMN
 490  CONTINUE
      WRITE (6,820) VEL(J), TOTRS2, SIGSOL
      DO 500 I = 1, N
        IWORK = NTTSE(I)
        IF (IWORK .GT. 20) IWORK = 20
        AMN = B(I,I) / NTTSE(I)
        AMN = SQRT(AMN) * TTEST(IWORK)
        WRITE (6,830) ASITE(I), E(I), EERR(I), EERG(I), F(I), FERR(I),
     1  FERG(I), TMTRM(I), TMTERR(I), TMTERG(I), TTSE(I), AMN,
     2  NTTSE(I), C(I,I)
 500  CONTINUE
C
C      WRITE OUT EQUATIONS THAT HAVE RESIDUALS> 3XSIGSOL
C
      WRITE (6,890)
      WRITE (7,890)
      WRITE (7,750) N, M, K, (VEL(I),VELERR(I),I=1,4)
      WRITE (7,660) (ASITE(I),I=1,N)
      IF (K) 530, 530, 510
 510  DO 520 I = 1, K
        WRITE (7,670) ALPHA(I), TTRM(I), TRERR(I), ASITE(NUM(I)),
     1  NUM(I)
 520  CONTINUE
 530  DO 540 I = 1, M
        RLIM = 3.0 * SIGSOL
        RMOD = ABS(RESID(I))
        IF (RMOD .LE. RLIM) WRITE (7,680) ALPHA1(I), ALPHA2(I), TT(I),
     1  TTERR(I), DIST(I), DISTER(I), ASITE(NUM1(I)), ASITE(NUM2(I)),
     2  NUM1(I), NUM2(I)
        IF (RMOD .GT. RLIM) WRITE (6,900) ASITE(NUM1(I)),
     1  ASITE(NUM2(I)), DIST(I), TT(I), RESID(I), RESERR(I)
 540  CONTINUE
C
C          ******* PLOT RESIDUALS ******
C
      CALL PAPER(1)
      YN = 0.0
      NDIM = 2000
      NOCHAR = 45
      CALL PLT(DIST, RESID, YN, DS, TS, M, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'DISTANCE (KM)', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, VEL(J), 2)
      CALL PLOTCS(6.0, 0.2, 'ST ERROR SOL=', 13)
      CALL PLOTNF(8.0, 0.2, SIGSOL, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'RESIDUALS', 9)
      CALL CTRORI(0.0)
C
C          ******* PLOT ANNOTATION ******
C
      IF (MEM .EQ. 1) GO TO 570
      MEM = 1
      CALL FTNCMD('ASSIGN 10=GPT9:XMAT-M25B', 24)
      READ (10,920) TITLE1
      READ (10,880) (KSTH(I),I=1,170)
      READ (10,910) (DIST25(I),I=1,170)
      DO 550 II = 1, N
        DO 550 JJ = 1, 170
 550  IF (ASITE(II) .EQ. KSTH(JJ)) X25(II) = DIST25(JJ)
      READ (5,630) NANOT
      READ (5,640) (ANOT(II),II=1,NANOT)
C
      DO 560 II = 1, NANOT
        DO 560 JJ = 1, 170
          IF (ANOT(II) .EQ. KSTH(JJ)) XANOT(II) = DIST25(JJ)
 560  CONTINUE
C
C          ******* PLOT TTERMS ******
C
 570  YN = -1.0
      NDIM = 200
      NOCHAR = 43
      CALL FRAME
      CALL PLT(X25, TMTRM, YN, DS, TS, N, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 580 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
 580  CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B (KM)', 23)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, VEL(J), 2)
      CALL PLOTCS(6.0, 0.2, 'ST ERROR SOL=', 13)
      CALL PLOTNF(8.0, 0.2, SIGSOL, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMNV, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'UNCORR TIME-TERMS', 17)
      CALL CTRORI(0.0)


C
C
C*************** CORRECT TTERMS FOR SHALLOWER STRUCTURE *************
C
C
      READ (5,650) IQ
```

```
      IF (IQ .EQ. 0) GO TO 610
      NDIM = 200
      CALL SHALL(TMTRM, ASITE, N, NDIM, VEL(J), VPG, VAVPG, TTRMPG, H)
      IF (NDIM .EQ. 0) GO TO 610
C
C           ******* PLOT CORRECTED TTERMS ******
C
      YN = -1.0
      NDIM = 200
      NOCHAR = 43
      CALL FRAME
      CALL PLT(X25, TMTRM, YN, DS, TS, N, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 590 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
 590  CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B (KM)', 23)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, VEL(J), 2)
      CALL PLOTCS(6.0, 0.2, 'ST ERROR SOL=', 13)
      CALL PLOTNF(8.0, 0.2, SIGSOL, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMNV, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, 'TIME-TERMS', 10)
      CALL CTRORI(0.0)
C
C           ******* PLOT CORRECTED DEPTHS ******
C
      YN = -2.0
      NDIM = 200
      NOCHAR = 43
      CALL FRAME
      CALL PLT(X25, H, YN, DS, TS, N, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 600 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
 600  CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B (KM)', 23)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, VEL(J), 2)
      CALL PLOTCS(6.0, 0.2, 'ST ERROR SOL=', 13)
      CALL PLOTNF(8.0, 0.2, SIGSOL, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMNV, 2)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.15, 3.0, '  DEPTHS  ', 10)
      CALL CTRORI(0.0)
      CALL FRAME
 610  CONTINUE
      CALL GREND


C
C
C********************** FORMAT STATEMENTS **************************
C
C
 620  FORMAT (I1)
 630  FORMAT (I2)
 640  FORMAT (18A4)
 650  FORMAT (I1)
 660  FORMAT ((4X,A4))
 670  FORMAT (4X, A4, 10X, 2F5.2, 6X, A4, I5)
 680  FORMAT (4X, A4, 6X, A4, 10X, 4F8.2, 2(4X,A4,2X), 2I5)
 690  FORMAT (//'*********** STATIONS WITH NO TT!!! ***********')
 700  FORMAT (1X, A4)
 710  FORMAT (/'DO YOU WANT TO USE THIS ROUTINE TO CALCULATE TTERMS?', /
     1        ' Y=1, N=0')
 720  FORMAT (//'CONSTRAINT STATION  ', A4, ' NOT FOUND IN INPUT LIST')
 730  FORMAT (//'COINCIDENT SHOT AND RECEIVER IN OBSERVATIONAL EQUATION
     1  ', I5, /'SHOT IS ', I5, '    STATION IS ', I5)
 740  FORMAT (//'EQUATION NUMBER ', I5, '  HAS A SHOT OR STATION NOT'/
     1        'GIVEN IN INITIAL INPUT LIST.'/'SHOT IS ', A8,
     2        '  STATION IS ', A8, /'POSITIONS IN INPUT LIST ARE ', 2I5,
     3        ' RESPECTIVELY')
 750  FORMAT (3(5X,I5), 8F5.2)
 760  FORMAT (' ', 'UNABLE TO CALCULATE LEAST SQUARES VELOCITY ')
 770  FORMAT ('0', 'LEAST SQUARE VELOCITY IS ', F5.2, ' +/- ', F5.2,
     1        ' +/- ', F5.2)
 780  FORMAT ('0', 'SUM OF D SQUARED IS ', F12.2, ' +/- ', F6.2,
     1        ' +/- ', F6.2)
 790  FORMAT (20A4)
 800  FORMAT (I5, 8F5.2)
 810  FORMAT ('0', 'STANDARD ERROR OF VELOCITY IS ', F8.4)
 820  FORMAT ('1', ' SOLUTION USING A VELOCITY OF ', F5.2, /,
     1        ' SUM OF RESIDULES SQUARED IS ', F5.2, /,
     2        ' STANDARD ERROR OF THE SOLUTION IS ', F5.2, ///,
     3
     4'STATION     E   ERROR  ERG      F   ERROR  ERG    TTERM ERROR  ER
     5G  B,W ST.ERR   95 CONF  NUM DATA')
 830  FORMAT (' ', A4, 1X, 3(3X,F6.2,2(1X,F5.2)), 2(5X,F5.2), 2I5)
 840  FORMAT (A8, 2X, 5012)
 850  FORMAT (//, 10X, '         5         10         15         20
     125         30         35         40         45         50')
```

```
 860 FORMAT (//, 10X, '          55         60         65         70
    175         80         85         90         95        100')
 870 FORMAT (//, 10X, '         105        110        115        120        1
    125        130        135        140        145        150')
 880 FORMAT (170A4)
 890 FORMAT (///'********* EQUATIONS WITH RESIDUALS><3XSIGSOL *******')
 900 FORMAT (2(1X,A4), 4F10.3)
 910 FORMAT (170F10.4)
 920 FORMAT (A4)
     STOP
     END
C
C
C******************** SUBROUTINE MATREAD ************************
C
C
C
     SUBROUTINE MATRD(ALPHA1, ALPHA2, ASITE, TT, TTERR, DIST, DISTER,
    1           M, N, VPG, VAVPG, TTRMPG)
     DIMENSION TT(2000), TTERR(2000), T(110,71), ET(110,71),
    1          ASTAT(110), ASHOT(110), ALPHA1(2000), ALPHA2(2000),
    2          XMAT(110,71), DIST(2000), DISTER(2000), XSTAT(110),
    3          XSHOT(110), ASITE(200), VAVPG(200), TTRMPG(200),
    4          VPG(200), SITE1(200), SITE2(200)
     REAL*4 N1, M2
     DATA PG /'PG  '/
     DATA PCP /'PCP '/
     DATA PMP /'PMP '/
     DATA PN /'PN  '/
     DATA PN1 /'PN1 '/
     DATA N1 /'N1  '/
     DATA S57 /'57  '/
     DATA M2 /'M2  '/
     DATA S2 /'2   '/

  10 WRITE (6,20)
  20 FORMAT (/'TRAVEL TIMES FOR WHICH PHASE? (EG. PG,PC1,PCP,PMP,PN)')
     READ (5,290) P
     WRITE (7,290) P
     IF (P .EQ. PG) GO TO 30
     IF (P .EQ. PCP) GO TO 40
     IF (P .EQ. PMP) GO TO 50
     IF (P .EQ. PN) GO TO 70
     IF (P .EQ. PN1) GO TO 80
  30 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
     GO TO 90
  40 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
     GO TO 90
  50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
     GO TO 90
  60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PM2P', 23)
     GO TO 90
  70 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
     GO TO 90
  80 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
     GO TO 90
  90 CONTINUE
     CALL FTNCMD('ASSIGN 8=GPT9:XMAT-CSSP', 23)
     CALL FTNCMD('ASSIGN 1=GPT9:XMAT-M25B', 23)
C
C     CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
C     IF NOT STOP
C
 100 READ (4,290) TIN
     IF (TIN .EQ. P) GO TO 120
     WRITE (6,110) TIN, P
 110 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
     STOP
C
C     READ IN VALUES ALREADY IN MATRIX FILE
C
 120 WRITE (6,280)
     READ (4,250) (ASTAT(I),I=1,110)
     READ (4,250) (ASHOT(I),I=1,71)
     READ (8,290) ADUMP
     READ (8,250) (XSTAT(I),I=1,110)
     READ (8,250) (XSHOT(I),I=1,71)
 130 DO 150 I = 1, 110
       READ (4,260) (T(I,J),ET(I,J),J=1,71)
       READ (8,260) (XMAT(I,J),J=1,71)
 140   IF (ASTAT(I) .EQ. XSTAT(I)) GO TO 150
       WRITE (6,270)
       STOP
 150 CONTINUE
     READ (5,160) NS
     READ (5,170) ((SITE1(I),SITE2(I)),I=1,NS)
     WRITE (6,180) ((SITE1(I),SITE2(I)),I=1,NS)
 160 FORMAT (I2)
 170 FORMAT ((2A4))
 180 FORMAT (('THE SITE ',A4,' IS MADE TO BE THE SAME AS SITE ',A4))
     DO 190 I = 1, 71
       DO 190 J = 1, NS
         IF (ASTAT(I) .EQ. SITE1(J)) ASTAT(I) = SITE2(J)
         IF (ASTAT(I) .EQ. SITE1(J)) XSTAT(I) = SITE2(J)
         IF (ASHOT(I) .EQ. SITE1(J)) ASHOT(I) = SITE2(J)
 190 IF (XSHOT(I) .EQ. SITE1(J)) XSHOT(I) = SITE2(J)
     READ (5,200) N
 200 FORMAT (I5)
     READ (5,210) ((ASITE(I),VPG(I),VAVPG(I),TTRMPG(I)),I=1,N)
 210 FORMAT ((A4,1X,3F10.3))
     L = 0
     DO 240 I = 1, 110
       DO 240 J = 1, 71
         IF (T(I,J) .EQ. 0.0) GO TO 230
         NST = 0
         NSH = 0
         DO 220 K = 1, N
           IF (ASTAT(I) .EQ. ASITE(K)) NST = K
 220       IF (ASHOT(J) .EQ. ASITE(K)) NSH = K
         IF (NST .EQ. 0 .OR. NSH .EQ. 0) GO TO 230
```

```
      L = L + 1
      TT(L) = T(I,J)
      TTERR(L) = ET(I,J)
      DIST(L) = XMAT(I,J)
      DISTER(L) = 0.05
      ALPHA1(L) = ASTAT(I)
      ALPHA2(L) = ASHOT(J)
230   IF (ASHOT(J) .EQ. XSHOT(J)) GO TO 240
      WRITE (6,270)
      STOP
240 CONTINUE
    M = L
250 FORMAT (110A4)
260 FORMAT (71(F10.4,F10.4))
270 FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
280 FORMAT (/'READING DATA FROM MATRIX FILE')
290 FORMAT (A4)
    RETURN
    END
C
C*****************************************************************
C
C
C ********************** PLT ***********************************
C
C
C*****************************************************************
C
C
    SUBROUTINE PLT(X, Y, YN, DS, TS, N, NDIM, NCCHAR)
C
C
    DIMENSION X(NDIM), Y(NDIM)
    XMAX = -9999.9
    XMIN = 9999.9
    YMAX = -9999.9
    YMIN = 9999.9
    DO 10 I = 1, N
      IF (X(I) .GT. XMAX) XMAX = X(I)
      IF (X(I) .LT. XMIN) XMIN = X(I)
      IF (Y(I) .GT. YMAX) YMAX = Y(I)
 10 IF (Y(I) .LT. YMIN) YMIN = Y(I)
    IF (YN .LT. 0.0) YMIN = 0.0
    IF (YN .EQ. - 2.0) XMAX = 400.0
    IF (YN .EQ. - 2.0.AND.XMIN.GE.0.0) XMIN =  0.0
    XMAX = XMAX + 0.1 * (XMAX - XMIN)
    XMIN = XMIN - 0.1 * (XMAX - XMIN)
    YMAX = YMAX + 0.1 * (YMAX - YMIN)
    YMIN = YMIN - 0.1 * (YMAX - YMIN)
    IF (YN .LT. 0.0) YMIN = 0.0
    IF (YN .EQ. - 2.0) YMAX = 50.0
    DS = 9.5 * 2.54
    TS = 6.0 * 2.54
    CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
    CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
    IF (YN .GE. 0) CALL MAP(XMIN, XMAX, YMIN, YMAX)
    IF (YN .LT. 0) CALL MAP(XMIN, XMAX, YMAX, YMIN)
    CALL CTRMAG(15)
    CALL SCALES
    CALL BORDER
    CALL PTPLOT(X, Y, 1, N, NCCHAR)
    CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
    CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
    CALL MAP(XMIN, XMAX, 0.0, 10.0)
    RETURN
    END
C
C*****************************************************************
C
C
C ********************  SHALL  ******************************
C
C
C*****************************************************************
C
C
    SUBROUTINE SHALL(TMTRM, ASITE, N, NDIM, VN, VPG, VAVPG, TTRMPG, H)
C
    DIMENSION TMTRM(NDIM), ASITE(NDIM), H(NDIM), VPG(NDIM),
   1          VAVPG(NDIM), TTRMPG(NDIM)
C
C
    IF (TTRMPG(1) .EQ. 0.0) NDIM = 0
    IF (NDIM .EQ. 0) RETURN
    WRITE (6,20)
    DO 10 I = 1, N
      VG = VPG(I)
      VC = VAVPG(I)
      IF (VN .LT. VC) STOP
      IF (VG .LT. VC) STOP
      IF (VN .LT. VG) STOP
      CORR = ((VG*(VN**2 - VC**2)**0.5 - VC*(VN**2 - VG**2)**0.5)/(VN*
   1  (VG**2 - VC**2)**0.5)) * TTRMPG(I)
      T = TMTRM(I) - CORR
      V = 6.37
      H(I) = T * VN * V / (VN**2 - V**2) ** 0.5
C      WRITE (6,30) ASITE(I), TMTRM(I), VC, VG, VN, CORR, TTRMPG(I), T,
C    1 H(I)
      TMTRM(I) = T
 10 CONTINUE
 20 FORMAT (//'OUTPUT FORM SUBROUTINE SHALL')
 30 FORMAT (1X, A4, 1X, 8F5.2)
    RETURN
    END
C
C
```

```
C
CA    AHJ Lewis
CD+
C*****************************************************************
C*                                                               *
C*        COMMON MID-POINT : TTERM PROGRAMME FOR REFLECTIONS     *
C*        ___________________________________________________    *
C*                 (uncorrected  travel times)                   *
C* AHJ LEWIS      Dept. Geol. Sciences, Durham          Nov 1985 *
C*****************************************************************
C
C
C                      Description
C                      ___________
C
C     This programme reads uncorrected travel times from the travel time
C     matrices. Programme reads distances of sites from M25B and sorts these
C     into CMP gathers. It then fits each CMP tt gather to the reflection
C     equation. The average velocity and depth are then calculated and
C     outputted to unit 6. The results are then rearranged for plotting.
C     The travel times are then corrected for shallow structure using apparent
C     velocities along common mid-point lines. The calculations are repeated
C     for these corrected travel times to produce an approximate measure of
C     the results corrected for shallow structure.
C
C     The reflection equation T**2 = X**2/V**2 + 4 * H**2 / V**2
C     is fitted to the travel-times for a wide-angle reflected phase
C     using the method suggested in the paper by
C           Bamford DAL, 1978  Interpretatioan of Wide-angle reflections
C                              in Realistic Earth-Mantle Structures
C                              GJRAS
C
C     Example input is provided in the file TTRMPMP.IN
C     The units are assigned as for the programme TTERMWAR.U
C
C
CD-
C
      REAL*8 A, B, SUMTT, SUMDD, C, D, TOTCD, TOTDD, V, TOTRS2, SETTRM
      DIMENSION NSTAT(400), NSHOT(400), ALPHA1(2000), ALPHA2(2000),
     1          T(2000), DIST(2000), TERR(2000), DISTER(2000),
     2          KSTH(180), DIST25(180), NCMP(2000),XCMP(400),TOTRSV(12),
     3          TCMP(2000), DCMP(2000), DMP(2000), SUMTT(400), VCON(12),
     4          SUMDD(400), TMAT(110,71), XMAT(110,71), ET(110,71),
     5          RESID(2000), D(2000), ASTAT(110), ASHOT(110), A(400),
     6          B(400), TTRM(400), C(2000), XANOT(20), ANOT(20),
     7          TITLE(20), H(400), TITLES(20), TTRMP(400), HP(400),
     8          XP(400), SETTRM(400), HERR(400), KSITE(400), VG(400),
     9          VAVPG(400), TTRMPG(400), TTEST(20), TRESID(2000)
      INTEGER ASTAT, ASHOT, BLANK, ALPHA1, ALPHA2, ANOT
      DATA M25B /'M25B'/
      DATA TTEST /12.7, 4.3, 3.18, 2.78, 5.57, 2.45, 2.37, 2.31, 2.26,
     1     2.23, 2.2, 2.18, 2.16, 2.15, 2.13, 2.12, 2.11, 2.10, 2.09,
     2     2.09/
C
C
C
C********************** READ IN DATA FROM MATRICES ***********************
C
C
      READ (5,440) TITLE
      WRITE (6,440) TITLE
      WRITE (7,440) TITLE
      CALL MATRD(ASTAT, ASHOT, TMAT, ET, XMAT)
      READ (5,20) NUMC, VADEF, HD, NSTCMP, NUM, (VCON(I),I=1,12)
      WRITE (7,20) NUMC, VADEF, HD, NSTCMP, NUM, (VCON(I),I=1,12)
      IF (NUMC .EQ. 0) WRITE (6,10)
   10 FORMAT (/'TT UNCORRECTED')
   20 FORMAT (I5, 2F5.2, 2I5, 12F5.2)
      IF (NUMC .EQ. 0) GO TO 30
      READ (5,470) ((KSITE(I),VG(I),VAVPG(I),TTRMPG(I)),I=1,NUMC)
      WRITE (7,470) ((KSITE(I),VG(I),VAVPG(I),TTRMPG(I)),I=1,NUMC)
   30 CALL FTNCMD('ASSIGN 10=GPT9:XMAT-M25B', 24)
      READ (10,440) TITLES
      READ (10,450) (KSTH(I),I=1,180)
      READ (10,460) (DIST25(I),I=1,180)
C
C
C                          **************
C*********************                    ****************************
C********************* LOOP OVER CMPS ****************************
C*********************                    ****************************
C                          **************
C
C
      MEMCOR = 0
   40 NTOT = 0
      TOTDD = 0.0
      TOTCD = 0.0
      TOTRS2 = 0.0
      DO 150 ICMP = 1, NUM
        IF (MEMCOR .EQ. 1) GO TO 100
C
C
C ************** FIND NEW XCMP
C
C
```

```
      SUMTT(IOMP) = 0.0
      SUMDD(IOMP) = 0.0
      XOMP(IOMP) = IOMP - 1.0 + NSTOMP
      WRITE (6,480) XOMP(IOMP)
C
C
C**************** FIND STATION SHOT PAIRS WITH XOMP(IOMP)
C
C     STATION SPACING IS 2KM
C     OMP SPACING IS 1KM <=> +/- 0.5KM !
C
 50   XU = XOMP(IOMP) + 0.5
      XD = XOMP(IOMP) - 0.5
C
C     LOOP OVER DATA TO FIND XOMPS
C
      NOMP(IOMP) = 0
      DO 90 I = 1, 180
        NSTRT = I + 1
        IF (I .EQ. 180) NSTRT = NSTRT - 1
        DO 80 J = NSTRT, 180
          XMP = 0.0
          IF (DIST25(I) .EQ. 0.0 .AND. KSTH(I) .NE. M25B)
     1        GO TO 80
          XMP = (DIST25(I) + DIST25(J)) / 2.0
C
C     IF XMP NOT EQUAL TO NEW XOMP TRY NEXT STAT/SHOT PAIR
C
          IF (XMP .EQ. 0.0) GO TO 80
          IF (XMP .GT. XU .OR. XMP .LT. XD) GO TO 80
C
C      FIND STAT/SHOT PAIR IN TMAT DATA
C
          IST = 0
          ISH = 0
          DO 60 IJ = 1, 110
            IF (KSTH(I) .EQ. ASTAT(IJ)) IST = IJ
            IF (KSTH(J) .EQ. ASTAT(IJ)) IST = IJ
            IF (IJ .GT. 71) GO TO 60
            IF (KSTH(J) .EQ. ASHOT(IJ)) ISH = IJ
            IF (KSTH(I) .EQ. ASHOT(IJ)) ISH = IJ
 60       CONTINUE
C
C        IF IST OR ISH IS ZERO THEN DEALING WITH 2 STAT/SHOTS
C        RESET L AND IST AND ISH
C
          IF (IST .NE. 0 .AND. ISH .NE. 0) GO TO 70
          IST = 0
          ISH = 0
          GO TO 90
C                 ------------------------------
C ************** FOUND STAT/SHOT PAIR WITH XMP=XOMP ******************
C                 ------------------------------
C.
 70       IF (TMAT(IST,ISH) .EQ. 0.0) GO TO 90
          NOMP(IOMP) = NOMP(IOMP) + 1
          NTOT = NTOT + 1
          ALPHA1(NTOT) = ASTAT(IST)
          ALPHA2(NTOT) = ASHOT(ISH)
          TOMP(NTOT) = TMAT(IST,ISH)
          DOMP(NTOT) = XMAT(IST,ISH)
          DMP(NTOT) = XMP
          SUMTT(IOMP) = SUMTT(IOMP) + TOMP(NTOT) ** 2
          SUMDD(IOMP) = SUMDD(IOMP) + DOMP(NTOT) ** 2
          WRITE (7,500) ASTAT(IST), ASHOT(ISH), DOMP(NTOT), TOMP(NTOT)
          GO TO 90
 80     CONTINUE
 90   CONTINUE
      GO TO 120
C              MISS OUT ON FIRST RUN THROUGH
C*********** CORRECTION FOR SHALLOW STRUCTURE ****************
C                      IF NUMC.NE.0
C                   IF NUMC.EQ.0 STOPS
C
 100  IF (NUMC .EQ. 0) GO TO 510
      IF (NOMP(IOMP) .LT. 2) GO TO 150
      N = NOMP(IOMP)
      CALL SHALL(ALPHA1, ALPHA2, TOMP, DOMP, N, NTOT, KSITE, VG,
     1      VAVPG, TTRMPG, NUMC, VADEF, HD)
      SUMTT(IOMP) = 0.0
      SUMDD(IOMP) = 0.0
      DO 110 J = 1, N
        NTOT = NTOT + 1
        SUMTT(IOMP) = SUMTT(IOMP) + TOMP(NTOT) ** 2
```

```
      SUMDD(IOMP) = SUMDD(IOMP) + DOMP(NTOT) ** 2
 110  CONTINUE
C
C
C                 ******************************
C****************                              ******************
C**************** CALCULATE LEAST SQUARES TOTALS ******************
C****************                              ******************
C****************      IF NOMP(IOMP)> OR = 2   ******************
C****************      IF NOT RESET VARIABLES  ******************
C****************                              ******************
C                 ******************************
C
C
 120  WRITE (6,370) NOMP(IOMP), NTOT
      IF (NOMP(IOMP) .GE. 2) GO TO 130
      IF (NOMP(IOMP) .EQ. 0) GO TO 150
      IF (MEMCOR .EQ. 1) GO TO 150
      TOMP(NTOT) = 0.0
      DOMP(NTOT) = 0.0
      NTOT = NTOT - 1
      GO TO 150
C
 130  B(IOMP) = SUMDD(IOMP) / NOMP(IOMP)
      A(IOMP) = SUMTT(IOMP) / NOMP(IOMP)
      N = NOMP(IOMP)
      ITOT = NTOT - N
      DO 140 IJ = 1, N
        ITOT = ITOT + 1
        D(ITOT) = DOMP(ITOT) ** 2 - B(IOMP)
        C(ITOT) = TOMP(ITOT) ** 2 - A(IOMP)
        TOTDD = TOTDD + D(ITOT) ** 2
 140  TOTCD = TOTCD + C(ITOT) * D(ITOT)
C      WRITE (6,270) TOTCD, TOTDD
C
C
      IF (ITOT .NE. NTOT) STOP
C
C
 150 CONTINUE
```

```
C
C                 ******************************
C****************                              ******************
C**************** CALCULATE LEAST SQUARES VELOCITY ******************
C****************                              ******************
C                 ******************************
C
C
      IF (TOTDD .GT. 0.0 .OR. TOTCD .GT. 0.0) GO TO 160
      WRITE (6,390)
      STOP
 160  VV = (TOTDD/TOTCD)
      V = VV ** 0.5
      WRITE (6,400) V
C
C
C********** CALCULATE ERRORS OVER THE XOMPS USED ******************
C
C
      DO 999 IV=1,12
      IF(IV.EQ.12) VCON(IV)= V
      IF(VCON(IV).EQ.0.0) GO TO 999
      VV= VCON(IV)**2
 169  L = 0
      ITOT = 0
      TOTRS2=0.0
      DO 180 I = 1, NUM
        IF (NOMP(I) .LT. 2) GO TO 180
        L = L + 1
        N = NOMP(I)
        SETTRM(I) = 0.0
        DO 170 J = 1, N
          ITOT = ITOT + 1
          RESID(ITOT) = C(ITOT) - D(ITOT) / VV
          SETTRM(I) = SETTRM(I) + RESID(ITOT) ** 2
          TOTRS2 = TOTRS2 + RESID(ITOT) ** 2
 170    CONTINUE
        TTRM(I) = A(I) - B(I) / VV
        DF = NOMP(I) * (NOMP(I) - 1)
        SETTRM(I) = SETTRM(I) / DF
        SETTRM(I) = (SETTRM(I)) ** 0.5
 180  CONTINUE
      TOTRSV(IV)= TOTRS2/10000.0
      WRITE(6,181) VCON(IV),TOTRS2, TOTRSV(IV)
 181  FORMAT(/' VELOCITY AND SUM RESIDUALS SQUARED ARE ', F10.3,F15.2,
     1F15.6)
 999  CONTINUE
      NUMP = L
      DF = NTOT - NUMP - 1
```

```
      SE = TOTRS2 / DF
      SE = SQRT(SE)
      SEV = SE * VV / TOTDD ** 0.5
      WRITE (6,410) V, SEV, SE



C
C
C********** CALCULATE TTRM AND DEPTHS FOR XOMPS ********************
C
C
      WRITE (6,280)
      L = 0
      DO 210 I = 1, NUM
        IF (NOMP(I) .LT. 2) GO TO 200
        IF (TTRM(I) .LT. 0.0) GO TO 190
        L = L + 1
        H(I) = TTRM(I) * VV / 4.0
        TTRM(I) = TTRM(I) ** 0.5
        H(I) = H(I) ** 0.5
        IF (MEMCOR .EQ. 1) H(I) = H(I) + HD
        HERR(I) = SETTRM(I) ** 2 + SEV ** 2
        SETTRM(I) = SETTRM(I) / 2.0
        HERR(I) = HERR(I) / 2.0
        IWORK = NOMP(I)
        IF (IWORK .GT. 20) IWORK = 20
        HERR95 = HERR(I) * TTEST(IWORK)
        TTRMP(L) = TTRM(I)
        HP(L) = H(I)
        XP(L) = XOMP(I)
        WRITE (6,290) XOMP(I), TTRM(I), SETTRM(I), H(I), HERR(I), HERR95
        GO TO 210
 190    TTRM(I) = NOMP(I)
        H(I) = NOMP(I)
        WRITE (6,310) XOMP(I)
 200    TTRM(I) = NOMP(I)
        H(I) = NOMP(I)
        WRITE (6,300) XOMP(I)
 210 CONTINUE
      NUMP = L


C
C
C********** WRITE OUT RESIDUALS ON EACH TRAVEL TIME USED **********
C
C
      WRITE (6,320)
      I = 0
      DO 230 ICMP = 1, NUM
        IF (NOMP(ICMP) .LT. 2) GO TO 230
        N = NOMP(ICMP)
        WRITE (6,480) XOMP(ICMP)
        DO 220 K = 1, N
          I = I + 1
          TSOL = TTRM(ICMP) ** 2 + DOMP(I) ** 2 / VV
          TSOL = TSOL ** 0.5
          TRESID(I) = TOMP(I) - TSOL
          WRITE (6,330) ALPHA1(I), ALPHA2(I), DOMP(I), TOMP(I),
     1    RESID(I), TRESID(I)
 220    CONTINUE
 230 CONTINUE


C
C
C          ******* PLOT RESIDUALS ******
C
      CALL PAPER(1)
      IF (MEMCOR .EQ. 1) CALL FRAME
      YN = 0.0
      NDIM = 2000
      NOCHAR = 45
      CALL PLT(DOMP, RESID, YN, DS, TS, NTOT, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'DISTANCE (KM)', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(6.0, 8.2, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.2, HD, 1)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, V, 2)
      CALL PLOTCS(6.0, 0.2, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.2, SE, 2)
      CALL CTRORI(1.0)
      IF (MEMCOR .EQ. 0) CALL PLOTCS(0.15, 3.0, 'RESIDUALS', 9)
      IF (MEMCOR .EQ. 1) CALL PLOTCS(0.15, 3.0, 'CORR RESIDUALS', 14)
      CALL CTRORI(0.0)
      CALL FRAME
      YN = 0.0
      NDIM = 2000
      NOCHAR = 45
      CALL PLT(DOMP, TRESID, YN, DS, TS, NTOT, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'DISTANCE (KM)', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(6.0, 8.2, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.2, HD, 1)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, V, 2)
      CALL PLOTCS(6.0, 0.2, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.2, SE, 2)
      CALL CTRORI(1.0)
```

```
      IF (MEMCOR .EQ. 0) CALL PLOTCS(0.15, 3.0, 'T,T RESIDUALS', 12)
      IF (MEMCOR .EQ. 1) CALL PLOTCS(0.15, 3.0, 'CORR TT  RESIDUALS',
     1     18)
      CALL CTRORI(0.0)


C
C******************** PLOT TOTRS2 AGAINST VELOCITY ******************
C
      CALL FRAME
      YN = 0.0
      NVCON=12
      NDIM = 2000
      NOCHAR = 45
      FMIN= TOTRSV(12)
      CALL PLT(VCON, TOTRSV, YN, DS, TS, NVCON, NDIM, NOCHAR)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(4.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTCS(1.0, 8.2, TITLE, 80)
      CALL PLOTCS(6.0, 8.2, 'NUM CMPS= ', 10)
      CALL PLOTNI(7.5, 8.2, NUMP)
      CALL PLOTCS(1.0, 0.2, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.2, V, 2)
      CALL PLOTCS(1.0, 1.3, 'FMIN = ', 7)
      CALL PLOTNF(2.2, 1.3, FMIN, 4)
      CALL PLOTCS(6.0, 0.2, 'NUM OF TT USED=   ', 18)
      CALL PLOTNI(8.0, 0.2, NTOT)
      CALL CTRORI(1.0)
      IF (MEMCOR .EQ. 0) CALL PLOTCS(0.15, 3.0, 'SUM RES 2', 9)
      IF (MEMCOR .EQ. 1) CALL PLOTCS(0.15, 3.0, 'SUM CORR RES 2', 14)
      CALL CTRORI(0.0)


C
C           ******* PLOT ANNOTATION ******
C
      IF (MEMCOR .EQ. 1) GO TO 250
      READ (5,340) NANOT
      READ (5,350) (ANOT(II),II=1,NANOT)
C
      DO 240 II = 1, NANOT
        DO 240 JJ = 1, 180
          IF (ANOT(II) .EQ. KSTH(JJ)) XANOT(II) = DIST25(JJ)
  240 CONTINUE



C
C           ******* PLOT TTERMS ******
C
  250 YN = -1.0
      NDIM = 200
      AMN = SEV
      NOCHAR = 43
      CALL FRAME
      CALL PLT(XP, TTRMP, YN, DS, TS, NUMP, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 260 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
  260 CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B OF CMP)', 26)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(6.0, 8.5, 'DATUM (KM)', 10)
      CALL PLOTNF(7.5, 8.5, HD, 1)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, V, 2)
      CALL PLOTCS(6.0, 0.32, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.32, SE, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMN, 4)
      CALL CTRORI(1.0)
      IF (MEMCOR .EQ. 0) CALL PLOTCS(0.15, 3.0, 'REFLECTION TIME-TERMS (
     1SECS)', 28)
      IF (MEMCOR .EQ. 1) CALL PLOTCS(0.15, 3.0, 'CORR REFLECT. TIME-TERM
     1S-DATUM', 30)
      CALL CTRORI(0.0)


C
C           ******* PLOT DEPTHS ******
C
      YN = -2.0
      NDIM = 200
      NOCHAR = 43
      CALL FRAME
      CALL PLT(XP, HP, YN, DS, TS, NUMP, NDIM, NOCHAR)
      CALL CTRORI(1.0)
      CALL CTRMAG(8)
      DO 270 II = 1, NANOT
        CALL PLOTNC(XANOT(II), 10.0, 43)
  270 CALL PLOTCS(XANOT(II), 10.1, ANOT(II), 4)
      CALL CTRORI(0.0)
      CALL CTRMAG(15)
      CALL PSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
      CALL MAP(0.0, 10.0, 0.0, 10.0)
      CALL PLOTCS(3.0, 8.5, 'DISTANCE FROM M25B OF CMP)', 26)
      CALL PLOTCS(1.0, 0.05, TITLE, 80)
      CALL PLOTCS(6.0, 8.5, 'DATUM (KM)', 10)
```

```
      CALL PLOTNF(7.5, 8.5, HD, 1)
      CALL PLOTCS(1.0, 0.32, 'VELOCITY KM/S', 13)
      CALL PLOTNF(2.9, 0.32, V, 2)
      CALL PLOTCS(3.5, 0.32, '+/-', 3)
      CALL PLOTNF(4.0, 0.32, AMN, 4)
      CALL PLOTCS(6.0, 0.32, 'ST. ERROR OF SOL =', 18)
      CALL PLOTNF(8.0, 0.32, SE, 2)
      CALL CTRORI(1.0)
      IF (MEMCOR .EQ. 0) CALL PLOTCS(0.15, 3.0, 'UNCORR DEPTH (KM)', 17)
      IF (MEMCOR .EQ. 1) CALL PLOTCS(0.15, 3.0, ' CORR DEPTH  (KM)', 17)
      CALL CTRORI(0.0)



C
C ********************** IF NUMC.NE.0 SET MEMCOR.EQ.1
C *******************
C                            IF MEMCOR.EQ.1 STOP
C
      IF (MEMCOR .EQ. 1 .OR. HD .EQ. 0.0) GO TO 510
      MEMCOR = 1
      IF (MEMCOR .EQ. 1 .AND. NUMC .NE. 0) GO TO 40

C
C*********************** FORMAT STATEMENTS ************************
C
C
C
  280 FORMAT (/////'TTERMS FOR THE OMPS USED'/'_____________________
     1_____'///
     2
     3'STAT SHOT   XCMP(M25B)  TTRM     ERROR     DEPTH+ERROR       95%
     4ERROR '/)
  290 FORMAT (1X, 6F10.3)
  300 FORMAT (1X, F10.3, ' THIS OMP NOT USED')
  310 FORMAT (1X, F10.3, ' THIS OMP TTERM<0!************')
  320 FORMAT (////'************ RESIDUALS ***************'//
     1        ' STAT  SHOT    DIST      TT          RESID '/)
  330 FORMAT (1X, A4, 1X, A4, 4F10.3)
  340 FORMAT (I2)
  350 FORMAT (20A4)
  360 FORMAT (1X, A4, '/', A4, F10.3)
  370 FORMAT (/' NUMBER= ', 2I5)
  380 FORMAT (/' TOTCD,TOTDD= ', 2F15.3)
  390 FORMAT (/'UNABLE TO CALCULATE LEAST SQUARES VELOCITY')
  400 FORMAT (/'LEAST SQUARE VELOCITY IS ', F10.3)
  410 FORMAT (//////' LEAST SQUARE VELOCITY = ', F5.3, '+/-', F7.4, /
     1        'STANDARD ERROR OF SOLUTION IS ', F10.4)
  420 FORMAT (I5)
  430 FORMAT ((20A4))
  440 FORMAT (20A4)
  450 FORMAT (3(60A4))
  460 FORMAT (3(60F10.4))
  470 FORMAT ((A4,1X,3F10.3))
  480 FORMAT (///'COMMON MID-POINT ', F10.3, /'_____________________
     1-')
  490 FORMAT (/'DID NOT FIND OMP')
  500 FORMAT (1X, A4, '/', A4, 2F10.3)


C                                  ____
C                        ********* END *********
C                                 -- --
  510 CALL GREND
      STOP
      END

C
C
C
C******************** SUBROUTINE MATREAD ************************
C
C
C
      SUBROUTINE MATRD(ASTAT, ASHOT, TMAT, ET, XMAT)
      DIMENSION TT(2000), TTERR(2000), TMAT(110,71), ET(110,71),
     1          ASTAT(110), ASHOT(110), XMAT(110,71), XSTAT(110),
     2          XSHOT(110)
      INTEGER XSHOT, XSTAT, ASTAT, ASHOT
      DATA PG /'PG  '/
      DATA PCP /'PCP '/
      DATA PMP /'PMP '/
      DATA PN /'PN  '/
      DATA PN1 /'PN1 '/

   10 WRITE (6,20)
   20 FORMAT (/'TRAVEL TIMES FOR WHICH PHASE? (EG. PG,PC1,PCP,PMP,PN)')
      READ (5,200) P
      WRITE (7,200) P
      IF (P .EQ. PG) GO TO 30
      IF (P .EQ. PCP) GO TO 40
      IF (P .EQ. PMP) GO TO 50
      IF (P .EQ. PN) GO TO 60
      IF (P .EQ. PN1) GO TO 70
   30 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PG', 21)
      GO TO 80
   40 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PCP', 22)
      GO TO 80
   50 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PMP', 22)
      GO TO 80
   60 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN', 21)
      GO TO 80
   70 CALL FTNCMD('ASSIGN 4=GPT9:TMAT-PN1', 22)
      GO TO 80
   80 CONTINUE
      CALL FTNCMD('ASSIGN 8=GPT9:XMAT-CSSP', 23)
C
C     CHECK CORRECT MATRIX FILE IS ASSIGNED TO UNIT 4
```

```
C     IF NOT STOP
C
  90 READ (4,200) TIN
     IF (TIN .EQ. P) GO TO 110
     WRITE (6,100) TIN, P
 100 FORMAT (/'INPUT TT-MAT IS ', A4, ' NOT ', A4)
     STOP
C
C     READ IN VALUES ALREADY IN MATRIX FILE
C
 110 WRITE (6,190)
     READ (4,160) (ASTAT(I),I=1,110)
     READ (4,160) (ASHOT(I),I=1,71)
     READ (8,200) ADUMP
     READ (8,160) (XSTAT(I),I=1,110)
     READ (8,160) (XSHOT(I),I=1,71)
 120 DO 140 I = 1, 110
       READ (4,170) (TMAT(I,J),ET(I,J),J=1,71)
       READ (8,170) (XMAT(I,J),J=1,71)
 130   IF (ASTAT(I) .EQ. XSTAT(I)) GO TO 140
       WRITE (6,180)
       STOP
 140 CONTINUE
 150 CONTINUE
 160 FORMAT (110A4)
 170 FORMAT (71(F10.4,F10.4))
 180 FORMAT (/'STAT/SHOT DO NOT CORRESPOND IN TMAT AND XMATS !!')
 190 FORMAT (/'READING DATA FROM MATRIX FILE')
 200 FORMAT (A4)
     RETURN
     END


C
C*****************************************************************
C
C
C ********************** PLT ***********************************
C
C
C*****************************************************************
C
C
     SUBROUTINE PLT(X, Y, YN, DS, TS, N, NDIM, NOCHAR)
C
C
     DIMENSION X(NDIM), Y(NDIM)
     XMAX = -99999999.9
     XMIN =  99999999.9
     YMAX = -99999999.9
     YMIN =  99999999.9
     DO 10 I = 1, N
       IF (X(I) .GT. XMAX) XMAX = X(I)
       IF (X(I) .LT. XMIN) XMIN = X(I)
       IF (Y(I) .GT. YMAX) YMAX = Y(I)
  10 IF (Y(I) .LT. YMIN) YMIN = Y(I)
     IF (YN .LT. 0.0) YMIN = 0.0
     IF (YN .EQ. - 2.0) XMAX = 400.0
     IF (YN .EQ. - 2.0 .AND. XMIN .GE. 0.0) XMIN = 0.0
     XMAX = XMAX + 0.1 * (XMAX - XMIN)
     XMIN = XMIN - 0.1 * (XMAX - XMIN)
     YMAX = YMAX + 0.1 * (YMAX - YMIN)
     YMIN = YMIN - 0.1 * (YMAX - YMIN)
     IF (YN .LT. 0.0) YMIN = 0.0
     IF (YN .EQ. - 2.0) YMAX = 50.0
     DS = 9.5 * 2.54
     TS = 6.0 * 2.54
     CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     IF (YN .GE. 0) CALL MAP(XMIN, XMAX, YMIN, YMAX)
     IF (YN .LT. 0) CALL MAP(XMIN, XMAX, YMAX, YMIN)
     CALL CTRMAG(15)
     CALL SCALES
     CALL BORDER
     CALL PTPLOT(X, Y, 1, N, NOCHAR)
     CALL PSPACE(0.1, DS/25.4 + 0.1, 0.1, TS/25.4 + 0.1)
     CALL CSPACE(0.0, DS/25.4 + 0.3, 0.0, TS/25.4 + 0.3)
     CALL MAP(XMIN, XMAX, 0.0, 10.0)
     RETURN
     END
C
C*****************************************************************
C
C
C
C*****************************************************************
C
C
C ******************** SHALL ********************************
C
C
C*****************************************************************
C
C
     SUBROUTINE SHALL(ALPHA1, ALPHA2, TOMP, DOMP, N, ITOT, KSITE, VPG,
    1          VAVPG, TTRMPG, NUMC, VADEF, HD)
C
     INTEGER ALPHA1, ALPHA2
     REAL*8 TT(400), X(400), RESULT(20)
     DIMENSION TMTRM(400), ALPHA1(2000), VPG(400), TOMP(2000),
    1          VAVPG(400), TTRMPG(400), ALPHA2(2000), DOMP(2000),
    2          KSITE(400), TUNC(2000), DUNC(2000)


C
C     REARRANGE EACH CMP IN DISTANCE ORDER
C
     N1 = N - 1
     DO 20 I = 1, N
```

```
      LL = 0
      L = ITOT
      DO 10 J = 1, N1
        L = L + 1
        IF (DOMP(L + 1) .GT. DOMP(L)) GO TO 10
        DTEMP = DOMP(L)
        DOMP(L) = DOMP(L + 1)
        DOMP(L + 1) = DTEMP
        TEMP = TOMP(L)
        TOMP(L) = TOMP(L + 1)
        TOMP(L + 1) = TEMP
        ITEMP = ALPHA1(L)
        ALPHA1(L) = ALPHA1(L + 1)
        ALPHA1(L + 1) = ITEMP
        ITEMP = ALPHA2(L)
        ALPHA2(L) = ALPHA2(L + 1)
        ALPHA2(L + 1) = ITEMP
   10   LL = 1
        IF (LL .EQ. 0) GO TO 30
   20 CONTINUE
   30 WRITE (7,40) N, ITOT
   40 FORMAT (/'OMP SORTED IN DISTANCE ORDER', 2I5)
      L = ITOT
      DO 50 I = 1, N
        L = L + 1
        TUNC(L) = TOMP(L)
   50 DUNC(L) = DOMP(L)
C
C
C************* FIND APPARENT VELOCITY FOR THIS OMP
C              IF N.LE.3 CANNOT USE MOVING AVERAGE
C              SO ASSIGN C THE GIVEN VELOCITY AND GOTO
C              DIFFERENT SECTION OF PROGRAMME
C
      IF (N .LE. 3) C = VADEF
      IF (N .LE. 3) GO TO 130


C
C     LOOP OVER ALL DATA AND CALCULATE VA FROM EACH 4 VALUES
C
      NSTRT = ITOT + 1
      NEND = ITOT + N - 3
      DO 120 L = NSTRT, NEND
        TT(1) = TUNC(L)
        TT(2) = TUNC(L + 1)
        TT(3) = TUNC(L + 2)
        TT(4) = TUNC(L + 3)
        X(1) = DUNC(L)
        X(2) = DUNC(L + 1)
        X(3) = DUNC(L + 2)
        X(4) = DUNC(L + 3)
        IFAIL = 0
        NN = 4
        CALL G02CAF(NN, X, TT, RESULT, IFAIL)
        IF (RESULT(6) .LE. 0.0) C = VADEF
        IF (RESULT(6) .LE. 0.0) GO TO 60
        C = 1 / RESULT(6)
        IF (N .EQ. 4) GO TO 130
   60   IF (C .LT. 4.0 .OR. C .GT. 12.0) C = VADEF
        WRITE (7,70) L, NSTRT, NEND, LL, C
   70   FORMAT (/'OUTPUT L, NSTRT, NEND,LL    '/, 4I5, /
     1          'APPARENT VELOCITY=', F5.2)


C
C     C  CALCULATED
C
        IF (L .EQ. NSTRT) LL = 0
   80   IF (L .GT. NSTRT) GO TO 90
        IF (LL .EQ. NSTRT) LL = L + 1
        IF (LL .EQ. 0) LL = L
        GO TO 100
   90   LL = LL + 1
C
C********** FIND EACH SHOT/STAT PAIR AND DO CORRECTION
C
  100   DO 110 J = 1, NUMC
          IF (ALPHA1(LL) .EQ. KSITE(J)) IST = J
  110   IF (ALPHA2(LL) .EQ. KSITE(J)) ISH = J
        IF (IST .EQ. 0 .OR. ISH .EQ. 0) STOP
        VG = VPG(IST)
        V1 = VAVPG(IST)
        IF (C .LE. V1) STOP
        IF (VG .LE. V1) STOP
        IF ((C - VG) .LT. 0.1) C = VG + 0.1
        H1 = TTRMPG(IST) * V1 * VG / (VG**2 - V1**2) ** 0.5
        TAN1 = V1 / (C**2 - V1**2) ** 0.5
        TAN2 = VG / (C**2 - VG**2) ** 0.5
        COS2 = (1 - VG**2/C**2) ** 0.5
        COS1 = (1 - V1**2/C**2) ** 0.5
        CORST = H1 * (1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
        CORSTX = H1 * (TAN1 - TAN2) + HD * TAN2
        VG = VPG(ISH)
        V1 = VAVPG(ISH)
        IF (C .LE. V1) STOP
        IF (VG .LE. V1) STOP
        IF ((C - VG) .LT. 0.1) C = VG + 0.1
        H1 = TTRMPG(ISH) * V1 * VG / (VG**2 - V1**2) ** 0.5
        TAN1 = V1 / (C**2 - V1**2) ** 0.5
        TAN2 = VG / (C**2 - VG**2) ** 0.5
        COS2 = (1 - VG**2/C**2) ** 0.5
        COS1 = (1 - V1**2/C**2) ** 0.5
        CORSH = H1 * (1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
        CORSHX = H1 * (TAN1 - TAN2) + HD * TAN2
        T = TOMP(LL) - CORST - CORSH
        D = DOMP(LL) - CORSTX - CORSHX
        WRITE (7,170) ALPHA1(LL), ALPHA2(LL), V1, VG, C, TTRMPG(ISH),
     1  H1, HD, CORST, CORSH, T, CORSTX, CORSHX, D
        TOMP(LL) = T
```

```
      DOMP(LL) = D
      IF (L .EQ. NSTRT .AND. LL .EQ. NSTRT) GO TO 80
      IF (LL .EQ. NEND + 1) GO TO 80
      IF (LL .EQ. NEND + 2) GO TO 80
 120 CONTINUE
     GO TO 180
C
C
C************* IF N.LE.3 READS C FROM FILE
C
C
C     LOOP OVER ALL DATA AND CALCULATE VA FROM EACH 4 VALUES
C
 130 IF (C .LT. 4.0 .OR. C .GT. 12.0) C = VADEF
     WRITE (7,160) C
     NSTRT = ITOT + 1
     NEND = ITOT + N
     DO 150 L = NSTRT, NEND
       LL = L
C
C********** FIND EACH SHOT/STAT PAIR AND DO CORRECTION
C
      DO 140 J = 1, NUMC
        IF (ALPHA1(LL) .EQ. KSITE(J)) IST = J
 140    IF (ALPHA2(LL) .EQ. KSITE(J)) ISH = J
      IF (IST .EQ. 0 .OR. ISH .EQ. 0) STOP
      VG = VPG(IST)
      V1 = VAVPG(IST)
      IF (C .LE. V1) STOP
      IF (VG .LE. V1) STOP
      IF ((C - VG) .LT. 0.1) C = VG + 0.1
      H1 = TTRMPG(IST) * V1 * VG / (VG**2 - V1**2) ** 0.5
      TAN1 = V1 / (C**2 - V1**2) ** 0.5
      TAN2 = VG / (C**2 - VG**2) ** 0.5
      COS2 = (1 - VG**2/C**2) ** 0.5
      COS1 = (1 - V1**2/C**2) ** 0.5
      CORST = H1 * (1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
      CORSTX = H1 * (TAN1 - TAN2) + HD * TAN2
      VG = VPG(ISH)
      V1 = VAVPG(ISH)
      IF (C .LT. V1) STOP
      IF (VG .LT. V1) STOP
      IF (C .LT. VG) C = VG
      H1 = TTRMPG(ISH) * V1 * VG / (VG**2 - V1**2) ** 0.5
      TAN1 = V1 / (C**2 - V1**2) ** 0.5
      TAN2 = VG / (C**2 - VG**2) ** 0.5
      COS2 = (1 - VG**2/C**2) ** 0.5
      COS1 = (1 - V1**2/C**2) ** 0.5
      CORSH = H1 * (1/(V1*COS1) - 1/(VG*COS2)) + HD / (VG*COS2)
      CORSHX = H1 * (TAN1 - TAN2) + HD * TAN2
      T = TOMP(LL) - CORST - CORSH
      D = DOMP(LL) - CORSTX - CORSHX
      WRITE (7,170) ALPHA1(LL), ALPHA2(LL), V1, VG, C, TTRMPG(ISH),
     1 H1, HD, CORST, CORSH, T, CORSTX, CORSHX, D
      TOMP(LL) = T
      DOMP(LL) = D
 150 CONTINUE

 160 FORMAT (/'OUTPUT FROM SUBROUTINE SHALL'/'APPARENT VELOCITY=',
     1       F5.2)
 170 FORMAT (2(1X,A4), 1X, 6(F5.2,1X), ' CORRS + TT    = ', 2F6.2,
     1       F10.4, /, 47X, ' CORRS + DIST  = ', 2F6.2, F10.4)
 180 RETURN
     END
C
C
C
```

```
C
CA    AHJ Lewis
CD+
C**************************************************************
C*                                                            *
C*                      CMPSYNSEI                             *
C*          ________________________________________          *
C*                                                            *
C* AHJ LEWIS      Dept. Geol. Sciences, Durham       Nov 1985 *
C**************************************************************
C
C
C                  Description
C                  ___________
C
C     THIS PROGRAMME CONTROLS CALLS TO SYNSEI IN ORDER TO
C     BUILD UP A COMMON SHOT/STATION REFRACTION STACK WHICH IS
C     WRITTEN TO UNIT 2 FOR PLOTTING USING THE CSSP PROGRAMME
C     CSSPLOT.U
C
C     EXAMPLE OF A RUN COMMAND:-
C     ___________________________________________________________
C     RUN *FTNX SCARDS=CMPSYNSEI
C     RUN -LOAD+SYNSEI+*GHOST 1=-1 2=-2 5=-5 6=-6 9=-9 10=,11=,12=,15=
C     ___________________________________________________________
C     UNIT  1 = OUTPUT OF BINARY SEISMOGRAM IN SYNSEI FORMAT
C     UNIT  2 = OUTPUT OF BINARY DATA IN I*2 PLOT FORMAT
C     UNIT  5 + EXAMPLE INPUT PROVIDED IN SYNSEI-CSSP IN GPT9:CSSP.DAT
C     UNIT  9 = OUTPUT OF PLOTFILES OF SELECTED TRACES
C     UNITS 6,10,11,12,15 USED BY SYNSEI-REFER TO DOCUMENTATION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      DIMENSION TILE(18),FILTIT(18),SHEAD(5),TITLES(18),ISTAT(50,2),
     1IEVENT(50,2),BALLA(50,1024),KSDAY(50),KSHOUR(50),KSMIN(50),
     2ABAL(50),JSAMP(50),DIST(50),SAMRAT(50)
      REAL*4 JDUR(50),KSSEC(50)
      INTEGER*2 LET,I2TEMP(1024)
      COMMON/RMCOP/ MINV,MAXV,NOV,LOV,MIND,MAXD,NOD,LOD,NOS,NOR
      COMMON/PMDEL/ ALL(50),ZZ(300)
      COMMON/VELR/ VRED
C
C
C
      READ(5,75) NUM
   75 FORMAT(I2)
      DO 76 K=1,NUM
        READ(5,80)(IEVENT(K,J),J=1,2),(ISTAT(K,J),J=1,2)
   80   FORMAT(2A4,2X,2A4)
   76   CONTINUE
C
C
C
C
      CALL SYNSEI(BALLA,JSAMP,DIST,ABAL,SAMRAT)
      DO 91 K=1,NUM
          WRITE(6,77) JSAMP(K),DIST(K),ABAL(K),SAMRAT(K)
   77     FORMAT(1H ,'JSAMP,DIST,ABAL,SAMRAT = ',I5,3F10.4)
         CALL STRTIM(ABAL(K),KSDAY(K),KSHOUR(K),KSMIN(K),KSSEC(K))
         JDUR(K)=SAMRAT(K)*(JSAMP(K)-1)
C
C
C      WRITE SYNTHETIC SEISMOGRAMS TO UNIT 1 WITH 1 LINE
C      HEADERS IN TAPEREAD FORMAT.
C
C
      IPOL=2
      GAIN=0.0012
      WRITE(2,90)(IEVENT(K,J),J=1,2),(ISTAT(K,J),J=1,2),KSDAY(K),
     1KSHOUR(K),KSMIN(K),KSSEC(K),JSAMP(K),SAMRAT(K),JDUR(K),GAIN,
     2IPOL,DIST(K)
   90 FORMAT(2A4,59X,2A4,38X,I4,1X,I2,2X,I2,F8.4,2X,I6,F10.4,F8.2,2X,
     1E10.4,2X,I4,F10.3)
      LET=JSAMP(K)*2
      N=JSAMP(K)
         DO 92 I=1,N
            I2TEMP(I)=IFIX(BALLA(K,I)*30000000.0)
   92       CONTINUE
         CALL WRITE(I2TEMP,LET,0,LNUM,2,&1000)
   91    CONTINUE
C
C
C      SEISMOGRAMS ARE NOW IN UNIT 2 : AVAILABLE FOR FURTHER
C      PLOTTING.
C
C
      GO TO 156
C
C
C      ERROR MESSAGES
C
C
 1000 WRITE(7,1010)
 1010 FORMAT(1X,'ERROR IN CALL WRITE')
      GO TO 156
  156 STOP
      END
C
C
C###########################################################
C
       SUBROUTINE STRTIM(TIME,KSDAY,KSHOUR,KSMIN,KSSEC)
C      ____________________________________________________
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       THIS SUBROUTINES CALCULATES TIME IN DAYS,HOURS,
C       MINUTES AND SECONDS GIVEN THE TIME IN DECIMAL
C       SECONDS
```

```
C
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      REAL KSSEC
        EE=0.0
        FF=0.0
        GG=0.0
        KSSEC=TIME
  61    IF(KSSEC.LT.60.0) GO TO 62
        KSSEC=KSSEC-60.0
        EE=1.0+EE
        GO TO 61
  62    KSMIN=EE
  63    IF(KSMIN.LT.60) GO TO 64
        KSMIN=KSMIN-60
        FF=1.0+FF
        GO TO 63
  64    KSHOUR=FF
  65    IF(KSHOUR.LT.24) GO TO 66
        KSHOUR=KSHOUR-24
        GG=1.0+GG
        GO TO 65
  66    KSDAY=GG
        RETURN
        END
C
```

```
C
C
C.......................                    ........................
C                       SEIS83 IMPLEMENTATION
C.......................                    ........................
C
```

All files are held on GPT9. All programmes are suffixed by S83.
The documentation is held in GPT9:SEIS83.DOC.
The following points should be noted:

1) only 146 interface points are allowed.
2) there is a time shift on the signal equal to
   0.45*gamma/freq
3) output in LU5 in S83SYNTPL is in the from which can
   be ploted using the programme CSSPLOT.

The most recent paper on SEIS83 is :

CERVENY V.,1985, Ray Synthetic seismograms for complex
two-dimensional and three-dimensional
stuctures.
J. Geophysics v58 2-26

It is recommended that this paper is read before using SEIS83.

FS tape list for tape CSPST1 on T3C1. Density= 6250 Volume=CSPST1 Lbltype=OS/VS Owner=GPT9 (14:19:21 on DEC 4, 1985)

File D	File Name	Ver#	Type	Lrecl	Size	Dev	Date and Time saved	Len(ft)	Userid
1	CST1N	1	Line	16012	117	Page	AUG. 20, 1985. 20:38:16	9	GLL4
2	CST2N	1	Line	16012	117	Page	AUG. 20, 1985. 20:40:32	16	GLL4
3	CST3N	1	Line	16012	117	Page	AUG. 20, 1985. 20:42:45	23	GLL4
4	CST4N	1	Line	16012	117	Page	AUG. 20, 1985. 20:45:25	31	GLL4
5	CST5N	1	Line	16012	117	Page	AUG. 20, 1985. 20:48:09	38	GLL4
6	CST6N	1	Line	16010	117	Page	AUG. 20, 1985. 20:51:54	46	GLL4
7	CST7N	1	Line	16010	117	Page	AUG. 20, 1985. 20:55:32	53	GLL4
8	CST8N	1	Line	16010	117	Page	AUG. 20, 1985. 20:59:22	61	GLL4
9	CST11N	1	Line	16096	93	Page	AUG. 20, 1985. 21:15:15	67	GLL4
10	CST19N	1	Line	16110	117	Page	AUG. 20, 1985. 21:20:04	75	GLL4
11	CST23N	1	Line	16110	117	Page	AUG. 20, 1985. 21:22:29	82	GLL4
12	CST25N	1	Line	16110	117	Page	AUG. 20, 1985. 21:25:17	89	GLL4
13	CST26N	1	Line	16110	117	Page	AUG. 20, 1985. 21:28:08	97	GLL4
14	CST27N	1	Line	16110	117	Page	AUG. 20, 1985. 21:31:34	104	GLL4
15	CST30N	1	Line	16046	117	Page	AUG. 20, 1985. 21:36:13	112	GLL4
16	CST31N	1	Line	16046	117	Page	AUG. 20, 1985. 21:39:46	119	GLL4
17	CST31NN	1	Line	16046	117	Page	AUG. 20, 1985. 21:42:26	127	GLL4
18	CST31SN	1	Line	16046	117	Page	AUG. 20, 1985. 21:45:00	134	GLL4
19	CST32N	1	Line	16046	117	Page	AUG. 20, 1985. 21:47:40	142	GLL4
20	CST34N	1	Line	15962	117	Page	AUG. 20, 1985. 21:50:43	149	GLL4
21	CST35N	1	Line	15962	117	Page	AUG. 20, 1985. 21:53:12	157	GLL4
22	CST36N	1	Line	15962	117	Page	AUG. 20, 1985. 21:55:29	164	GLL4
23	CST37N	1	Line	15962	117	Page	AUG. 20, 1985. 21:57:35	172	GLL4
24	CST42N	1	Line	15950	117	Page	AUG. 20, 1985. 22:01:14	179	GLL4
25	CST44N	1	Line	15950	117	Page	AUG. 21, 1985. 17:54:22	187	GLL4
26	CST45N	1	Line	15950	117	Page	AUG. 21, 1985. 17:58:38	194	GLL4
27	CST46N	1	Line	15950	117	Page	AUG. 21, 1985. 18:02:57	202	GLL4
28	CST47N	1	Line	15950	117	Page	AUG. 21, 1985. 18:06:53	209	GLL4
29	CST51N	1	Line	15950	117	Page	AUG. 21, 1985. 18:10:35	217	GLL4
30	CST38N	1	Line	15962	117	Page	AUG. 21, 1985. 18:16:55	224	GLL4
31	CST39N	1	Line	15962	117	Page	AUG. 21, 1985. 18:19:35	231	GLL4
32	CST40N	1	Line	15962	117	Page	AUG. 21, 1985. 18:21:55	239	GLL4
33	CST41N	1	Line	15962	117	Page	AUG. 21, 1985. 18:24:10	246	GLL4
34	CST48N	1	Line	16060	117	Page	AUG. 21, 1985. 18:27:02	254	GLL4
35	CST49N	1	Line	16060	117	Page	AUG. 21, 1985. 18:30:01	261	GLL4
36	CST50N	1	Line	16060	117	Page	AUG. 21, 1985. 18:33:18	269	GLL4
37	CST53N	1	Line	16060	117	Page	AUG. 21, 1985. 18:36:14	276	GLL4
38	CST52N	1	Line	16748	117	Page	AUG. 21, 1985. 18:44:18	284	GLL4
39	CST33N	1	Line	15324	111	Page	AUG. 21, 1985. 18:46:43	291	GLL4

FS tape list for tape CSPST1 on T3C1. Density= 6250 Volume=CSPST1 Lbltype=OS/VS Owner=GPT9 (14:19:21 on DEC 4, 1985)

File D	File Name	Ver#	Type	Lrecl	Size	Dev	Date and Time saved	Len(ft)	Userid
1	CST1N	1	Line	16012	117	Page	AUG. 20, 1985. 20:38:16	9	GLL4
2	CST2N	1	Line	16012	117	Page	AUG. 20, 1985. 20:40:32	16	GLL4
3	CST3N	1	Line	16012	117	Page	AUG. 20, 1985. 20:42:45	23	GLL4
4	CST4N	1	Line	16012	117	Page	AUG. 20, 1985. 20:45:25	31	GLL4
5	CST5N	1	Line	16012	117	Page	AUG. 20, 1985. 20:48:09	38	GLL4
6	CST6N	1	Line	16010	117	Page	AUG. 20, 1985. 20:51:54	46	GLL4
7	CST7N	1	Line	16010	117	Page	AUG. 20, 1985. 20:55:32	53	GLL4
8	CST8N	1	Line	16010	117	Page	AUG. 20, 1985. 20:59:22	61	GLL4
9	CST11N	1	Line	16096	93	Page	AUG. 20, 1985. 21:15:15	67	GLL4
10	CST19N	1	Line	16110	117	Page	AUG. 20, 1985. 21:20:04	75	GLL4
11	CST23N	1	Line	16110	117	Page	AUG. 20, 1985. 21:22:29	82	GLL4
12	CST25N	1	Line	16110	117	Page	AUG. 20, 1985. 21:25:17	89	GLL4
13	CST26N	1	Line	16110	117	Page	AUG. 20, 1985. 21:28:08	97	GLL4
14	CST27N	1	Line	16110	117	Page	AUG. 20, 1985. 21:31:34	104	GLL4
15	CST30N	1	Line	16046	117	Page	AUG. 20, 1985. 21:36:13	112	GLL4
16	CST31N	1	Line	16046	117	Page	AUG. 20, 1985. 21:39:46	119	GLL4
17	CST31NN	1	Line	16046	117	Page	AUG. 20, 1985. 21:42:26	127	GLL4
18	CST31SN	1	Line	16046	117	Page	AUG. 20, 1985. 21:45:00	134	GLL4
19	CST32N	1	Line	16046	117	Page	AUG. 20, 1985. 21:47:40	142	GLL4
20	CST34N	1	Line	15962	117	Page	AUG. 20, 1985. 21:50:43	149	GLL4
21	CST35N	1	Line	15962	117	Page	AUG. 20, 1985. 21:53:12	157	GLL4
22	CST36N	1	Line	15962	117	Page	AUG. 20, 1985. 21:55:29	164	GLL4
23	CST37N	1	Line	15962	117	Page	AUG. 20, 1985. 21:57:35	172	GLL4
24	CST42N	1	Line	15950	117	Page	AUG. 20, 1985. 22:01:14	179	GLL4
25	CST44N	1	Line	15950	117	Page	AUG. 21, 1985. 17:54:22	187	GLL4
26	CST45N	1	Line	15950	117	Page	AUG. 21, 1985. 17:58:38	194	GLL4
27	CST46N	1	Line	15950	117	Page	AUG. 21, 1985. 18:02:57	202	GLL4
28	CST47N	1	Line	15950	117	Page	AUG. 21, 1985. 18:06:53	209	GLL4
29	CST51N	1	Line	15950	117	Page	AUG. 21, 1985. 18:10:35	217	GLL4
30	CST38N	1	Line	15962	117	Page	AUG. 21, 1985. 18:16:55	224	GLL4
31	CST39N	1	Line	15962	117	Page	AUG. 21, 1985. 18:19:35	231	GLL4
32	CST40N	1	Line	15962	117	Page	AUG. 21, 1985. 18:21:55	239	GLL4
33	CST41N	1	Line	15962	117	Page	AUG. 21, 1985. 18:24:10	246	GLL4
34	CST48N	1	Line	16060	117	Page	AUG. 21, 1985. 18:27:02	254	GLL4
35	CST49N	1	Line	16060	117	Page	AUG. 21, 1985. 18:30:01	261	GLL4
36	CST50N	1	Line	16060	117	Page	AUG. 21, 1985. 18:33:18	269	GLL4
37	CST53N	1	Line	16060	117	Page	AUG. 21, 1985. 18:36:14	276	GLL4
38	CST52N	1	Line	16748	117	Page	AUG. 21, 1985. 18:44:18	284	GLL4
39	CST33N	1	Line	15324	111	Page	AUG. 21, 1985. 18:46:43	291	GLL4

40	CST16N	1	Line	16612	116	Page	AUG.	21,	1985.	18:49:46	298	GLL4
41	CST54N	1	Line	16046	117	Page	AUG.	21,	1985.	19:00:11	306	GLL4
42	CST55N	1	Line	16046	117	Page	AUG.	21,	1985.	19:02:26	313	GLL4
43	CST56N	1	Line	16046	117	Page	AUG.	21,	1985.	19:04:47	321	GLL4
44	CST57N	1	Line	16046	117	Page	AUG.	21,	1985.	19:07:55	328	GLL4
45	CST59N	1	Line	16046	117	Page	AUG.	21,	1985.	19:10:55	336	GLL4
46	CST60N	1	Line	16046	117	Page	AUG.	21,	1985.	19:13:57	343	GLL4
47	CST58N	1	Line	16046	117	Page	AUG.	21,	1985.	19:17:03	351	GLL4
48	CST1M	1	Line	15990	101	Page	AUG.	21,	1985.	19:22:44	357	GLL4
49	CST2M	1	Line	15990	101	Page	AUG.	21,	1985.	19:25:17	364	GLL4
50	CST3M	1	Line	15990	101	Page	AUG.	21,	1985.	19:27:46	370	GLL4
51	CST4M	1	Line	15990	101	Page	AUG.	21,	1985.	19:30:14	377	GLL4
52	CST5M	1	Line	15990	101	Page	AUG.	21,	1985.	19:32:44	384	GLL4
53	CST6M	1	Line	15990	101	Page	AUG.	21,	1985.	19:36:13	390	GLL4
54	CST7M	1	Line	15990	101	Page	AUG.	21,	1985.	19:39:59	397	GLL4
55	CST8M	1	Line	15990	101	Page	AUG.	21,	1985.	19:44:06	403	GLL4
56	CST13M	1	Line	16086	101	Page	AUG.	21,	1985.	19:50:12	410	GLL4
57	CST15M	1	Line	16086	101	Page	AUG.	21,	1985.	19:53:12	417	GLL4
58	CST19M	1	Line	16086	93	Page	AUG.	21,	1985.	19:56:14	423	GLL4
59	CST21M	1	Line	16086	97	Page	AUG.	21,	1985.	19:58:48	429	GLL4
60	CST23M	1	Line	16086	101	Page	AUG.	21,	1985.	20:06:54	436	GLL4
61	CST25M	1	Line	16086	101	Page	AUG.	21,	1985.	20:09:23	442	GLL4
62	CST27M	1	Line	16086	101	Page	AUG.	21,	1985.	20:11:51	449	GLL4
63	CST30M	1	Line	16018	101	Page	AUG.	21,	1985.	20:15:35	456	GLL4
64	CST31M	1	Line	16018	101	Page	AUG.	21,	1985.	20:19:18	462	GLL4
65	CST31NM	1	Line	16020	101	Page	AUG.	21,	1985.	20:27:13	469	GLL4
66	CST32M	1	Line	16020	101	Page	AUG.	21,	1985.	20:31:54	475	GLL4
67	CST34M	1	Line	15950	101	Page	AUG.	21,	1985.	20:35:40	482	GLL4
68	CST35M	1	Line	15950	101	Page	AUG.	21,	1985.	20:38:49	489	GLL4
69	CST36M	1	Line	15950	101	Page	AUG.	21,	1985.	20:42:02	495	GLL4
70	CST37M	1	Line	15950	101	Page	AUG.	21,	1985.	20:45:11	502	GLL4
71	CST38M	1	Line	15948	101	Page	AUG.	21,	1985.	20:47:48	508	GLL4
72	CST39M	1	Line	15948	101	Page	AUG.	21,	1985.	20:50:11	515	GLL4
73	CST40M	1	Line	15948	101	Page	AUG.	21,	1985.	20:52:36	522	GLL4
74	CST42M	1	Line	15936	101	Page	AUG.	21,	1985.	21:01:58	528	GLL4
75	CST44M	1	Line	15936	101	Page	AUG.	21,	1985.	21:05:50	535	GLL4
76	CST45M	1	Line	15936	101	Page	AUG.	21,	1985.	21:08:35	541	GLL4
77	CST46M	1	Line	15936	101	Page	AUG.	21,	1985.	21:11:23	548	GLL4
78	CST47M	1	Line	15936	101	Page	AUG.	21,	1985.	21:14:18	555	GLL4
79	CST51M	1	Line	15936	101	Page	AUG.	21,	1985.	21:19:26	561	GLL4

80	CST54M	1	Line	16028	101	Page	AUG.	21,	1985.	21:36:39	568	GLL4
81	CST57M	1	Line	16028	101	Page	AUG.	21,	1985.	21:40:28	574	GLL4
82	CST60M	1	Line	16028	101	Page	AUG.	21,	1985.	21:44:19	581	GLL4
83	CST11M	1	Line	15786	89	Page	AUG.	21,	1985.	21:49:34	587	GLL4
84	CST11M	2	Line	0	1	Page	AUG.	22,	1985.	10:41:15	588	GLL4
85	CST49M	1	Line	16050	89	Page	AUG.	22,	1985.	10:47:45	594	GLL4
86	CST50M	1	Line	16050	89	Page	AUG.	22,	1985.	10:52:10	600	GLL4
87	CST53M	1	Line	16050	89	Page	AUG.	22,	1985.	10:58:48	606	GLL4
88	CST48M	1	Line	16050	89	Page	AUG.	22,	1985.	11:04:22	612	GLL4
89	CST55M	1	Line	16028	85	Page	AUG.	22,	1985.	11:13:03	618	GLL4
90	CST56M	1	Line	16028	97	Page	AUG.	22,	1985.	11:16:42	624	GLL4
91	CST33M	1	Line	15640	101	Page	AUG.	22,	1985.	11:21:42	631	GLL4
92	CST11N	2	Line	16096	93	Page	SEP.	10,	1985.	18:44:02	637	GLL4
93	CST52M	1	Line	0	1	Page	SEP.	20,	1985.	11:55:30	638	GLL4
94	CST52M	2	Line	16990	98	Page	SEP.	20,	1985.	13:29:28	644	GLL4
95	CST54M	2	Line	16028	97	Page	OCT.	30,	1985.	12:13:14	651	GLL4
96	CST60M	2	Line	16028	97	Page	OCT.	30,	1985.	12:13:19	657	GLL4
97	CST11N	3	Line	0	1	Page	NOV.	01,	1985.	14:08:29	658	GLL4
98	CST11N	4	Line	16096	93	Page	NOV.	01,	1985.	14:18:20	664	GLL4
99	CST11M	3	Line	15786	89	Page	NOV.	05,	1985.	12:19:23	670	GLL4
100	CST/R5	1	Line	7930	48	Page	DEC.	04,	1985.	14:18:38	674	GLL4

FS tape list for tape CSPSH2 on T3C1. Density= 6250 Volume=CSPSH2 Lbltype=OS/VS Owner=GPT9 (14:26:16 on DEC 4, 1985)

File D	File Name	Ver#	Type	Lrecl	Size	Dev	Date and Time saved	Len(ft)	Userid
1	CSH/N25	1	Line	16654	213	Page	AUG. 21, 1985. 17:49:30	14	GPT9
2	CSH/M25A	1	Line	16904	201	Page	AUG. 21, 1985. 17:50:11	26	GPT9
3	CSH/N16	1	Line	16652	217	Page	AUG. 22, 1985. 19:00:20	39	GPT9
4	CSH/M16	1	Line	16908	201	Page	AUG. 22, 1985. 19:01:06	51	GPT9
5	CSH/N12	1	Line	16624	214	Page	AUG. 23, 1985. 10:14:12	64	GPT9
6	CSH/M12	1	Line	16930	150	Page	AUG. 23, 1985. 10:14:44	73	GPT9
7	CSH/N11	1	Line	16650	217	Page	AUG. 23, 1985. 10:15:28	86	GPT9
8	CSH/M11	1	Line	16940	213	Page	AUG. 23, 1985. 10:16:04	99	GPT9
9	CSH/N1	1	Line	16686	214	Page	AUG. 23, 1985. 21:39:30	111	GPT9
10	CSH/M1	1	Line	16080	197	Page	AUG. 23, 1985. 21:40:10	123	GPT9
11	CSH/N2	1	Line	16646	214	Page	AUG. 23, 1985. 21:40:45	136	GPT9
12	CSH/M3	1	Line	16084	197	Page	AUG. 23, 1985. 21:41:47	148	GPT9
13	CSH/M2	1	Line	16080	197	Page	SEP. 04, 1985. 21:41:02	160	GPT9
14	CSH/N9	1	Line	16626	217	Page	SEP. 04, 1985. 22:23:12	173	GPT9
15	CSH/I1B	1	Line	16763	194	Page	SEP. 04, 1985. 22:23:37	184	GPT9
16	CSH/N13	1	Line	16628	213	Page	SEP. 04, 1985. 22:23:57	197	GPT9
17	CSH/N23B	1	Line	16620	214	Page	SEP. 04, 1985. 22:24:20	210	GPT9
18	CSH/M25B	1	Line	16904	201	Page	SEP. 04, 1985. 22:24:40	222	GPT9
19	CSH/N23A	1	Line	16674	221	Page	SEP. 04, 1985. 22:25:02	235	GPT9
20	CSH/M17	1	Line	16896	185	Page	SEP. 04, 1985. 22:25:29	246	GPT9
21	CSH/N19	1	Line	16670	214	Page	SEP. 04, 1985. 22:25:48	259	GPT9
22	CSH/M20	1	Line	16884	201	Page	SEP. 04, 1985. 22:26:12	271	GPT9
23	CSH/M21	1	Line	16882	201	Page	SEP. 04, 1985. 22:27:15	283	GPT9
24	CSH/M14	1	Line	16914	201	Page	SEP. 04, 1985. 22:27:34	295	GPT9
25	CSH/M8	1	Line	16546	201	Page	SEP. 04, 1985. 22:56:33	307	GPT9
26	CSH/N6	1	Line	16748	217	Page	SEP. 04, 1985. 22:57:02	320	GPT9
27	CSH/M4	1	Line	16624	201	Page	SEP. 04, 1985. 22:57:25	332	GPT9
28	CSH/N3	1	Line	16674	214	Page	SEP. 04, 1985. 22:57:45	345	GPT9
29	CSH/M1	2	Line	16080	197	Page	SEP. 04, 1985. 22:58:13	357	GPT9
30	CSH/N28	1	Line	16672	217	Page	SEP. 04, 1985. 22:58:42	370	GPT9
31	CSH/N15	1	Line	16650	217	Page	SEP. 04, 1985. 22:59:03	383	GPT9
32	CSH/M10	1	Line	16954	198	Page	SEP. 04, 1985. 22:59:34	395	GPT9
33	CSH/N21	1	Line	16682	213	Page	SEP. 04, 1985. 23:00:02	408	GPT9
34	CSH/N8	1	Line	16634	217	Page	SEP. 05, 1985. 01:34:37	421	GPT9
35	CSH/N7	1	Line	16692	217	Page	SEP. 05, 1985. 01:35:15	433	GPT9
36	CSH/N5	1	Line	16692	217	Page	SEP. 05, 1985. 01:35:58	446	GPT9
37	CSH/N4	1	Line	16714	217	Page	SEP. 05, 1985. 01:39:49	459	GPT9
38	CSH/M5	1	Line	16612	201	Page	SEP. 05, 1985. 02:45:27	471	GPT9
39	CSH/M6	1	Line	16552	197	Page	SEP. 05, 1985. 02:53:03	483	GPT9

40	CSH/M7	1	Line	16582	194	Page	SEP.	05,	1985.	02:53:46	495	GPT9
41	CSH/M9	1	Line	16990	201	Page	SEP.	05,	1985.	02:54:21	507	GPT9
42	CSH/N10	1	Line	16640	217	Page	SEP.	05,	1985.	04:16:16	520	GPT9
43	CSH/N14	1	Line	16650	217	Page	SEP.	05,	1985.	04:16:51	533	GPT9
44	CSH/N17	1	Line	16660	217	Page	SEP.	05,	1985.	04:17:15	546	GPT9
45	CSH/N18	1	Line	16644	217	Page	SEP.	05,	1985.	04:17:37	559	GPT9
46	CSH/M13	1	Line	16016	37	Page	SEP.	05,	1985.	15:22:28	562	GPT9
47	CSH/M15	1	Line	16920	201	Page	SEP.	05,	1985.	15:22:43	574	GPT9
48	CSH/M18	1	Line	16900	185	Page	SEP.	05,	1985.	15:23:25	585	GPT9
49	CSH/M19	1	Line	16908	201	Page	SEP.	05,	1985.	15:24:15	597	GPT9
50	CSH/N22	1	Line	16646	217	Page	SEP.	06,	1985.	13:49:06	610	GLL4
51	CSH/N24	1	Line	16664	217	Page	SEP.	06,	1985.	13:49:30	623	GLL4
52	CSH/N26	1	Line	16644	217	Page	SEP.	06,	1985.	13:49:52	636	GLL4
53	CSH/N27	1	Line	16678	217	Page	SEP.	06,	1985.	13:50:19	649	GLL4
54	CSH/N29	1	Line	16660	213	Page	SEP.	06,	1985.	13:50:44	662	GLL4
55	CSH/I1A	1	Line	16872	198	Page	SEP.	06,	1985.	13:57:29	674	GLL4
56	CSH/M22	1	Line	16896	201	Page	SEP.	06,	1985.	14:05:40	686	GLL4
57	CSH/M23	1	Line	16874	201	Page	SEP.	06,	1985.	14:06:04	698	GLL4
58	CSH/M24	1	Line	16894	201	Page	SEP.	06,	1985.	14:06:24	710	GLL4
59	CSH/N20	1	Line	16654	217	Page	SEP.	06,	1985.	21:04:09	723	GLL4
60	CSH/M6	2	Line	16552	197	Page	SEP.	16,	1985.	10:09:20	735	GLL4
61	CSH/M7	2	Line	16582	194	Page	SEP.	16,	1985.	10:28:43	746	GLL4
62	CSH/M11	2	Line	16940	201	Page	SEP.	19,	1985.	17:41:43	758	GLL4
63	CSH/M13	2	Line	16084	193	Page	SEP.	19,	1985.	17:43:07	770	GLL4
64	CSH/N23A	2	Line	16674	217	Page	SEP.	19,	1985.	18:37:07	783	GLL4
65	CSH/N24	2	Line	16364	213	Page	SEP.	19,	1985.	18:45:07	796	GLL4
66	CSH/M18	2	Line	16900	181	Page	SEP.	19,	1985.	18:46:06	807	GLL4
67	CSH/M20	2	Line	16884	193	Page	SEP.	19,	1985.	18:55:22	818	GLL4
68	CSH/M15	2	Line	16920	197	Page	SEP.	19,	1985.	18:55:42	830	GLL4
69	CSH/I1A	2	Line	16872	190	Page	SEP.	19,	1985.	18:56:07	842	GLL4
70	CSH/I1B	2	Line	16763	190	Page	SEP.	19,	1985.	19:01:51	853	GLL4
71	CSH/M11	3	Line	16940	197	Page	SEP.	19,	1985.	19:02:08	865	GLL4
72	CSH/M12	2	Line	16930	185	Page	SEP.	20,	1985.	18:25:55	876	GLL4
73	CSH/M17	2	Line	16896	177	Page	SEP.	20,	1985.	18:30:26	887	GLL4
74	CSH/N7	2	Line	16692	217	Page	OCT.	21,	1985.	13:30:21	900	GLL4
75	CSH/N6	2	Line	16748	217	Page	OCT.	21,	1985.	13:36:00	913	GLL4
76	CSH/N4	2	Line	16714	217	Page	OCT.	21,	1985.	13:40:18	926	GLL4
77	CSH/M1	3	Line	16080	197	Page	OCT.	24,	1985.	18:18:46	938	GLL4
78	CSH/M1	4	Line	16080	197	Page	OCT.	24,	1985.	18:28:24	949	GLL4

FS tape list for tape CSPCD1 on T3C1. Density= 6250 Volume=CSPCD1 Lbltype=OS/VS Owner=GPT9 (14:37:29 on DEC 4, 1985)

File D	File Name	Ver#	Type	Lrecl	Size	Dev	Date and Time saved	Len(ft)	Userid
1	CDS/94.7	1	Line	16086	149	Page	MAY 28, 1985. 20:26:12	10	GLL4
2	CDS/131.92	1	Line	16094	197	Page	MAY 28, 1985. 20:26:37	22	GLL4
3	CDS/146.92	1	Line	16660	189	Page	MAY 28, 1985. 20:31:54	34	GLL4
4	CDS/180.07	1	Line	16110	161	Page	MAY 28, 1985. 20:33:35	43	GLL4
5	CDS/98.84	1	Line	16644	169	Page	MAY 28, 1985. 20:34:55	54	GLL4
6	CDS/193.16	1	Line	16108	133	Page	SEP. 19, 1985. 18:05:41	62	GLL4
7	CDS/167.08	1	Line	16110	133	Page	SEP. 19, 1985. 18:09:15	70	GLL4
8	CDS/154.63	1	Line	16110	165	Page	SEP. 19, 1985. 18:12:08	81	GLL4
9	CDS/134.11	1	Line	16644	177	Page	SEP. 19, 1985. 18:15:11	91	GLL4
10	CDS/124.47	1	Line	16664	145	Page	SEP. 19, 1985. 18:17:48	100	GLL4
11	CDS/116.53	1	Line	16612	157	Page	SEP. 19, 1985. 18:20:17	110	GLL4
12	CDS/107.34	1	Line	16654	146	Page	SEP. 19, 1985. 18:22:51	119	GLL4
13	CDS/103.01	1	Line	16670	138	Page	SEP. 19, 1985. 18:25:17	128	GLL4
14	CDS/85.68	1	Line	16086	121	Page	SEP. 19, 1985. 18:27:18	135	GLL4
15	CDS/64.23	1	Line	16640	70	Page	SEP. 19, 1985. 18:29:18	140	GLL4
16	CSH/85.68	1	Line	16086	117	Page	SEP. 20, 1985. 20:35:20	148	GLL4
17	CSH/94.7	1	Line	16086	141	Page	SEP. 20, 1985. 20:40:56	157	GLL4
18	CDS/98.84	2	Line	16644	161	Page	SEP. 20, 1985. 20:44:17	166	GLL4
19	CDS/131.92	2	Line	16094	185	Page	SEP. 20, 1985. 20:48:49	178	GLL4
20	CDS/107.34	2	Line	16654	142	Page	SEP. 20, 1985. 21:01:35	187	GLL4
21	CDS/124.47	2	Line	16090	137	Page	SEP. 20, 1985. 21:06:09	195	GLL4
22	CDS/134.11	2	Line	16644	169	Page	SEP. 20, 1985. 21:48:00	205	GLL4
23	CDS/180.07	2	Line	16110	141	Page	SEP. 23, 1985. 18:46:08	214	GLL4
24	CDS/193.16	2	Line	16108	125	Page	SEP. 23, 1985. 18:48:25	222	GLL4
25	CDS/154.63	2	Line	16110	161	Page	SEP. 23, 1985. 18:50:59	232	GLL4
26	CDS/146.92	2	Line	16660	177	Page	SEP. 23, 1985. 18:53:13	243	GLL4
27	CDS/134.11	3	Line	16644	165	Page	SEP. 23, 1985. 18:55:27	253	GLL4
28	CDS/131.92	3	Line	16094	181	Page	SEP. 23, 1985. 18:58:28	264	GLL4
29	CDS/116.53	2	Line	16612	149	Page	SEP. 23, 1985. 19:00:47	273	GLL4
30	CDS/107.34	3	Line	16654	134	Page	SEP. 23, 1985. 19:03:46	281	GLL4
31	CDS/103.01	2	Line	16670	134	Page	SEP. 23, 1985. 19:05:38	290	GLL4
32	CDS/98.84	3	Line	16644	157	Page	SEP. 23, 1985. 19:07:27	300	GLL4
33	CDS/94.7	2	Line	16086	141	Page	SEP. 23, 1985. 19:11:33	308	GLL4
34	CDS/85.68	2	Line	16086	117	Page	SEP. 23, 1985. 19:15:34	316	GLL4
35	CDS/85.68	3	Line	16086	113	Page	SEP. 23, 1985. 19:24:14	323	GLL4
36	CDS/94.7	3	Line	16086	105	Page	SEP. 23, 1985. 19:24:24	330	GLL4
37	CDS/94.7	4	Line	16086	105	Page	SEP. 23, 1985. 19:57:50	337	GLL4

FS tape list for tape CSPMP1 on T3C0. Density= 6250 Volume=CSPMP1 Lbltype=OS/VS Owner=GPT9 (13:53:27 on DEC 9, 1985)

File D	File Name	Ver#	Type	Lrecl	Size	Dev	Date and Time saved	Len(ft)	Userid
1	CMP30/M1	1	Line	16554	62	Page	DEC. 06, 1985. 20:43:50	6	GLL4
2	CMP40/N1	1	Line	16044	61	Page	DEC. 06, 1985. 20:44:07	10	GLL4
3	CMP60/N1	1	Line	16692	102	Page	DEC. 06, 1985. 20:45:09	17	GLL4
4	CMP1/M1	1	Line	16872	82	Page	DEC. 06, 1985. 20:45:30	22	GLL4
5	CMP60/N10	1	Line	16650	70	Page	DEC. 09, 1985. 13:52:03	27	GLL4
6	CMP1/M15	1	Line	16084	33	Page	DEC. 09, 1985. 13:52:58	30	GLL4

Appendix E

CSSP NATURE PAPER - BOTT ET AL 1985

Reprinted from Nature, Vol. 314, No. 6013, pp. 724-727, 25 April 1985
© *Macmillan Journals Ltd., 1985*

Crustal structure south of the Iapetus suture beneath northern England

M. H. P. Bott, R. E. Long, A. S. P. Green*, A. H. J. Lewis, M. C. Sinha* & D. L. Stevenson

Department of Geological Sciences, University of Durham, Durham DH1 3LE, UK

The crustal structure just south of the Iapetus suture[1] along a line extending from the mid-North Sea high across northern England to the Solway Basin has been investigated as part of the Caledonian Suture Seismic Experiment of 1982. This unique sea-to-land wide-angle explosion seismology project has produced data of unprecedented quantity and quality. The main innovation was the use of both closely spaced shots (mainly at sea) and closely spaced stations (mainly on land), yielding a data set which can eventually be arranged in a variety of new ways for phase correlation and modelling, such as constant offset and common mid-point. The line was extended by Irish and German groups across Ireland to the Shannon Estuary, having a total length of over 600 km, but here we present only preliminary results for northern England and adjacent marine regions. The results for this region differ radically from those of the earlier LISPB experiment[2] in that we have recognized a well-defined 6.15 km s^{-1} basement interpreted as pre-Caledonian metamorphic basement beneath most of the line at relatively shallow depth, and there is a well-defined mid-crustal refractor starting at approximately 16 km depth beneath the North Sea and northern England which is not recognized beneath the Irish Sea.

The following four main phases have been clearly recognized from the North Sea shots recorded in northern England, as illustrated by the recordings of the shots at station S32 (Figs 1, 2): (1) The first arrival beyond about 30 km is the P_g phase which has a velocity of 6.15 ± 0.02 m s^{-1} at 2-4 km depth as determined by time-term analysis. [The symbol P refers to compressional waves. P_g travels in the upper crust. P_cP and P_mP are wide-angle reflections or diving waves from respectively the mid-crustal discontinuity/gradient and the Moho. P_n is the Moho head wave refracted at the crust/mantle boundary.] P_g consistently dies out at all stations for shots beyond about N11-N13. Such decay of the P_g arrival has previously been cited as evidence for low-velocity zones in the upper crust. In the absence of laterally displaced sections containing similar offset distances, it has not been possible in past experiments to distinguish satisfactorily between the effects of a low-velocity zone and effects of lateral velocity variation in the upper crust. In this case, however, P_g is seen to decay in the vicinity of a specific shot position irrespective of station, so the cause must be interpreted as a lateral inhomogeneity near that shot position rather than as an underlying low-velocity zone. (2) A well-defined P_cP phase of apparent velocity ~6.5 km s^{-1} results from a steep increase in velocity with depth of ~6.22-6.5 km s^{-1} at 16-18 km depth, also detected in some earlier work[3,4]. (3) The P_mP phase representing the wide-angle reflection from the Moho is a clear large-amplitude arrival. (4) A small amplitude P_n phase of apparent velocity 7.9 km s^{-1} occurs as the first arrival beyond the crossover distance of 130 km. The P_mP and P_n phases are consistent with a sharp Moho at 30 km depth.

A model of the crustal structure assuming horizontal layering beneath the variable shallow structure, together with a synthetic seismogram computed using the reflectivity method[5], is shown in Fig. 2.

The Irish Sea shots recorded in northern England (Fig. 3) yielded three main phases: (1) A P_g phase with a velocity of 6.15 ± 0.02 km s^{-1} determined by time-term analysis occurs as the first arrival between about 30 km and the crossover distance of 128 km, and comes from a depth of 2-4 km. Arrivals from shots over the Permo-Triassic basin beneath the Solway Firth are systematically delayed by the low-velocity sediments. (2) A conspicuous second arrival of 6.8-7.0 km s^{-1} apparent velocity is interpreted as the diving wave from a velocity gradient above the Moho at 28-32 km depth. (3) A P_n phase with an apparent velocity of 8.3 km s^{-1} or higher occurs as the first arrival beyond 128 km. The high apparent velocity is attributed to laterally varying time delays caused by the shallow sediment structure of the Solway basin.

A model of the velocity-depth distribution consistent with these three phases is shown in Fig. 3. A steep increase in velocity marking the top of a lower crustal layer, such as is indicated by the North Sea shots, is not apparent in the Irish Sea. Nevertheless, the possible existence of a higher-velocity lower crust, possibly underlying a velocity inversion at shallower levels, cannot be ruled out at this stage.

The shallow structure along the line of the experiment, which is probably just south of the Iapetus suture, is shown down to the 6.15 km s^{-1} basement in Fig. 4*a*. The structure down to the base of the Permo-Triassic was not resolved by the experiment and has been obtained from published sources[6] and from seismic reflection data supplied by Shell. The depth to the 5.5 km s^{-1}

* Present addresses: Geothermal Energy Project, Rosemanowes Quarry, Herniss, Penryn, Cornwall, UK (A.S.P.G.); Department of Earth Sciences, Bullard Laboratories, Madingley Rise, Madingley Road, Cambridge, UK (M.C.S.)

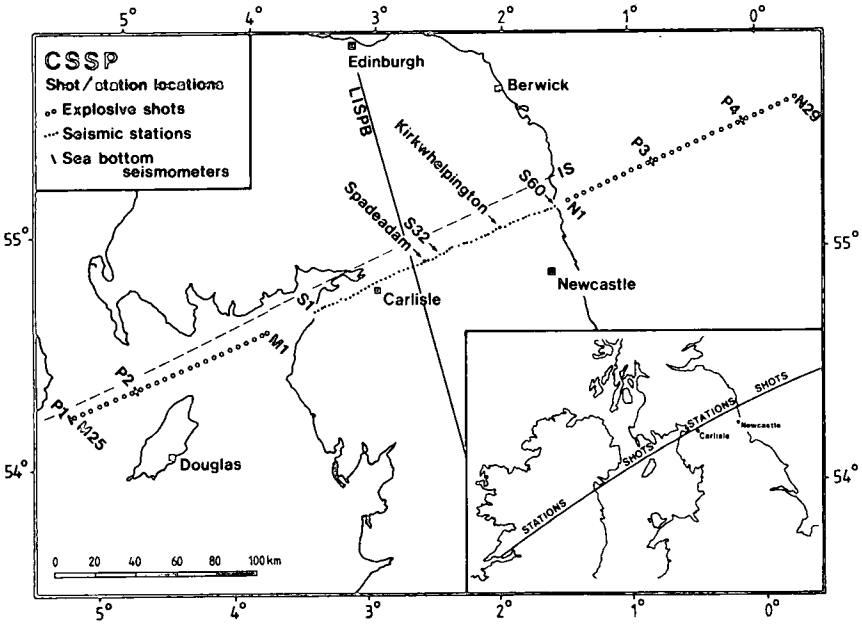


Fig. 1 Location of Caledonian Suture Seismic Project (CSSP), with details for Irish Sea to North Sea, showing also the supposed Iapetus suture[1] (IS) and the LISPB line[2].

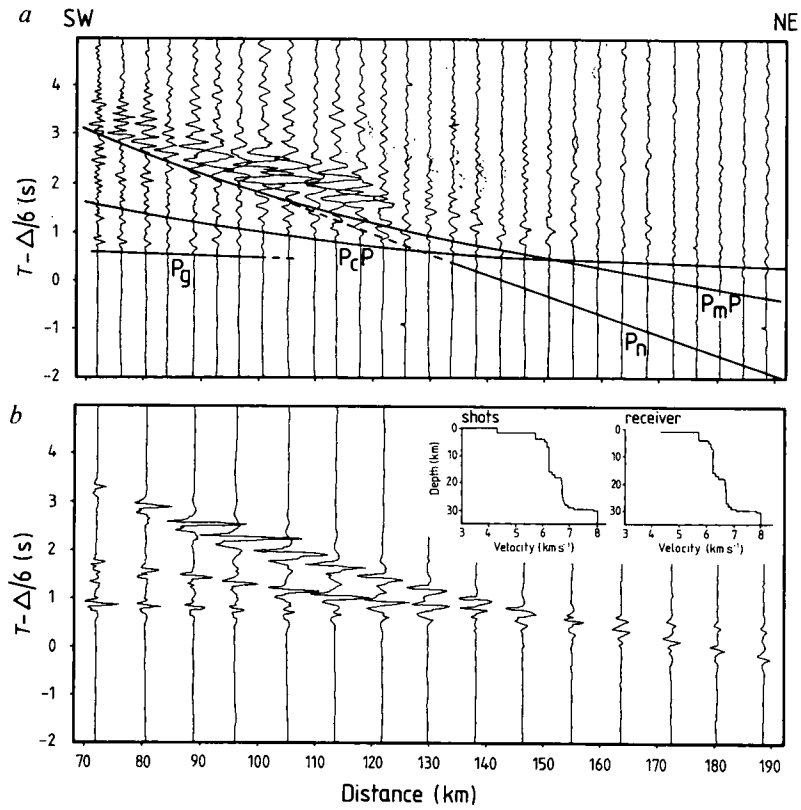


Fig. 2 *a*, Seismic recordings of North Sea shots N1–N29 at station S32 (Fig. 1) reduced to a velocity of 6.0 km s^{-1}, showing main phases computed for the synthetic model. *b*, Synthetic seismogram interpretation of *a* with inset velocity-depth distributions.

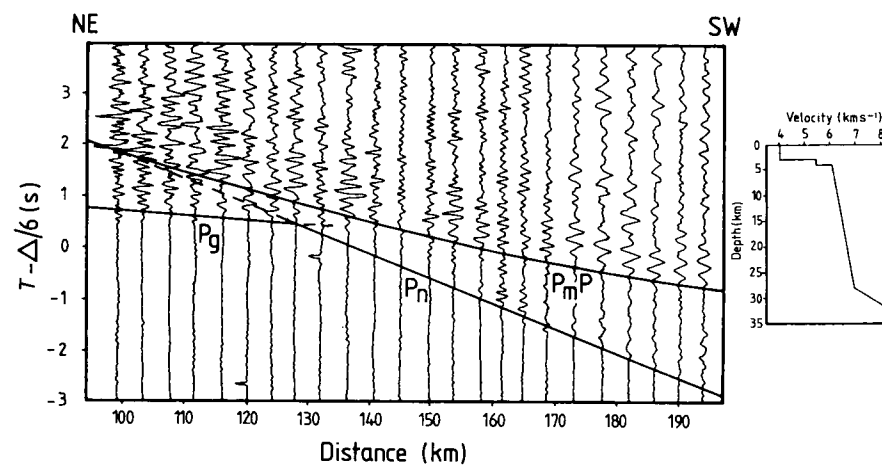


Fig. 3 Seismic recordings of Irish Sea shots M2–M25 at station S32 (Fig. 1) reduced to 6.0 km s^{-1}, showing the main phases computed by ray tracing for inset velocity-depth model.

Lower Palaeozoic layer has been determined from first arrivals out to ~30 km range. The 6.15 km s^{-1} basement is well defined and its depth, as shown, has been inferred from time-term analysis.

The recognition of this 6.15 km s^{-1} upper crustal refractor at the relatively shallow depth of 4 km or less is a result of considerable significance. This layer extends down to about 16 km depth beneath the North Sea and Northumberland Trough, but its base is undetermined beneath the Irish Sea. The evidence indicates a slight increase in velocity with depth without obvious indication of a low-velocity layer within it. According to Hall *et al*[7,8], velocities >6.0 km s^{-1} at shallow depth beneath the Southern Uplands (north of the suture) represent crystalline metamorphic basement beneath the Lower Palaeozoic sequence rather than a metamorphic transition to greenschist facies within the Lower Palaeozoic[9]. Recordings along the Northumberland Trough of a land shot at Spadeadam (Fig. 1) indicate that the boundary between the 5.5–5.7 km s^{-1} Lower Palaeozoic rocks underlying the Carboniferous and the 6.15 km s^{-1} layer is sharp rather than gradational, which appears to rule out a metamorphic transition within the Lower Palaeozoic. The 6.15 km s^{-1} velocity is too low for oceanic crust. Both the velocity and the aeromagnetic anomalies[10] along the line are probably too uniform for island-arc-type crust. The simplest interpretation is that the Lower Palaeozoic rocks are relatively thin just south of the Iapetus suture, where they are underlain by crystalline basement which is probably formed of Precambrian metamorphic rocks. The contact could be an unconformity or a tectonic feature. This interpretation differs radically from the LISPB profile[2] crossing our line at Spadeadam, which indicated a 5.8–6.0 km s^{-1} Lower Palaeozoic basement extending down to at least 15 km depth.

A second important result is that the deep crust appears to differ between the North Sea/Northumberland Trough region and the Irish Sea region. P_cP has not been unambiguously detected beneath the Irish Sea, in contrast to the excellent North Sea development, and the Moho is interpreted as being gradational beneath the Irish Sea but relatively sharp beneath the North Sea. Taken at face value, this indicates juxtaposition of different crustal types along strike just south of the suture. This

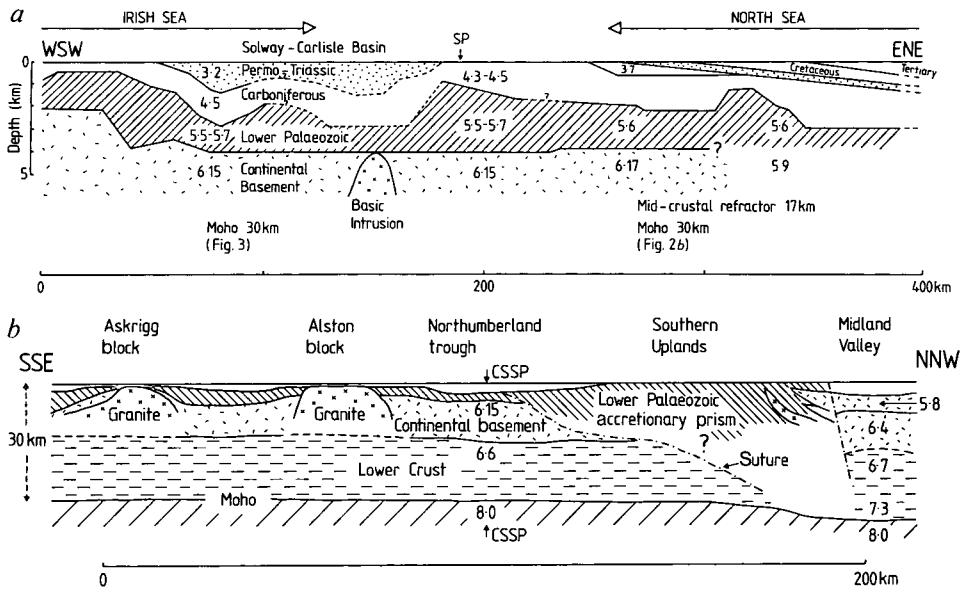


Fig. 4 *a*, Shallow crustal structure along the line of the experiment (Fig. 1), showing also the approximate locations of the crustal models of Figs 2*b* and 3. SP, Spadeadam. The basic intrusion beneath the Carlisle basin is inferred from the aeromagnetic map[10]. Seismic velocities in km s^{-1}. *b*, Speculative cross-section perpendicularly crossing the line of the experiment (marked CSSP) from the Midland Valley to the Northern Pennines, showing the possible relationship of the crust south of the Iapetus suture to the Southern Uplands accretionary prism. Seismic velocities are shown in km s^{-1} for this experiment (CSSP) and for the Midland Valley (LISPB[2]).

conclusion, however, should be treated with some caution. The low-frequency P_cP and P_mP phases (Fig. 2) are most conspicuously developed for the shots in deeper water in the North Sea, which have significant energy in the 3-5-Hz band. It is possible that the lack of detection of P_cP for the Irish Sea shots, and the weaker P_mP development, may stem from the observed lack of significant energy in this band from the shallower Irish Sea shots. Future work aims to distinguish between these possibilities and to study transition between the two crustal types.

Comparison of our results with the relevant part of the WINCH crustal seismic reflection line[11] down the west of Britain is of interest. The WINCH line crosses the western extremity of our Irish Sea shot line, and a short section of WINCH was shot along the western end of our line to allow comparison between the vertical incidence and wide-angle surveys. A conspicuous feature of the WINCH data in the vicinity of our line is the concentration of strong sub-parallel and almost horizontal reflectors which occur at two-way travel times of 5-10 s, contrasting with the much more transparent regions above and below. The lower limit of these reflectors appears to correspond with the Moho, but from the present preliminary interpretation we have not been able to identify a velocity-depth signature beneath the Irish Sea corresponding to the upper boundary of the zone. It is possible that the distinctive 6.5 km s^{-1} lower crust at the North Sea end of our line corresponds to this zone of reflectors, which is well developed elsewhere on the mainland of England[12], but this needs future testing.

In conclusion, Fig. 4*b* is a speculative section showing our interpretation of how the crustal structure inferred from this experiment may relate to the Southern Uplands accretionary prism[13] to the north and the deep structure of the Pennine region[4] to the south. The velocity-depth distribution beneath the Midland Valley comes from LISPB[2] and the Iapetus suture itself has been inserted in the form suggested by the WINCH line[11] further west. The relationship of the Lower Palaeozoic accretionary prism to the overthrust southern crust explains the observation of a 6.5 km s^{-1} lower crust beneath Eskdalemuir in the Southern Uplands[3] and why a higher velocity lower crust was not detected by LISPB[2].

We thank NERC, the Royal Society, the University of Durham, Mobil and Esso for research grants, Shell for a Studentship (A.H.J.L.) and NERC for shiptime and loan of seismological field equipment. We also thank those who gave field assistance during the project.

Received 6 December 1984; accepted 12 February 1985.

1. Phillips, W. E. A., Stillman, C. J. & Murphy, T. *Jl geol. Soc. Lond.* **132**, 579-609 (1976).
2. Bamford, D., Nann, K., Prodehl, C. & Jacob, B. *Geophys. J. R. astr. Soc.* **54**, 43-60 (1978).
3. Jacob, A. W. B. *Geophys. J. R. astr. Soc.* **18**, 189-197 (1969).
4. Bott, M. H. P., Swinburn, P. M. & Long, R. E. *Proc. Yorks. geol. Soc.* **44**, 479-495 (1984).
5. Fuchs, K. & Müller, G. *Geophys. J. R. astr. Soc.* **23**, 417-433 (1971).
6. Day, G. A. *et al.* in *Petroleum Geology of the Continental Shelf of North-West Europe* (eds Illing, L. V. & Hobson, G. D.) 76-84 (Institute of Petroleum, London, 1981).
7. Hall, J. *et al. Nature* **305**, 418-420 (1983).
8. Hall, J. *et al. Nature* **309**, 89-90 (1984).
9. Oliver, G. J. H. & McKerrow, W. S. *Nature* **309**, 89 (1984).
10. *1:250 000 Series Aeromagnetic Anomaly Maps* (Institute of Geological Sciences, London).
11. Brewer, J. A. *et al. Nature* **305**, 206-210 (1983).
12. Whittaker, A. & Chadwick, R. A. *Geol. Mag.* **121**, 621-624 (1984).
13. Leggett, J. K., McKerrow, W. S. & Eales, M. H. *Jl geol. Soc. Lond.* **136**, 755-770 (1979).

DEPARTMENT OF
GEOLOGICAL SCIENCES
PUBLICATION No. 829
SCIENCE LABORATORIES
THE UNIVERSITY, DURHAM

Printed in Great Britain by Macmillan Production Ltd., Basingstoke, Hampshire