

JAERI-Data/Code

96-001



H_∞反応度推定法実証のためのVHTRC実験

1996年2月

藤井義雄・鈴木勝男・秋濃藤義・山根 剛
藤崎伸吾・竹内素允・小野俊彦

日本原子力研究所
Japan Atomic Energy Research Institute

本レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合せは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）あて、お申し越しください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1996

編集兼発行 日本原子力研究所

印 刷 (株)原子力資料サービス

H ∞ 反応度推定法実証のための VHTRC 実験

日本原子力研究所東海研究所原子炉工学部

藤井 義雄・鈴木 勝男・秋濃 藤義・山根 剛
藤崎 伸吾・竹内 素允・小野 俊彦

(1996年1月5日受理)

この実験は、H ∞ 反応度推定法を実証するため VHTRC においてデータ収集を行ったものである。本報告書では、実験の方法、測定回路およびデータ処理ソフトウェアの詳細について述べる。

VHTRC Experiment for Verification Test of
 H^∞ Reactivity Estimation Method

Yoshio FUJII, Katsuo SUZUKI, Fujiyoshi AKINO
Tsuyoshi YAMANE, Shingo FUJISAKI, Motoyoshi TAKEUCHI
and Toshihiko ONO

Department of Reactor Engineering
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received January 5, 1996)

This experiment was performed at VHTRC to acquire the data for verifying the H^∞ reactivity estimation method. In this report, the experimental method, the measuring circuits and data processing softwares are described in details.

Keywords: Reactivity Estimation, Estimator, H^∞ Control, VHTRC

目 次

1. はじめに	1
2. VHTRC の概要	2
2.1 原子炉集合体	2
2.2 運転条件	3
3. 実験	12
3.1 実験の概要	12
3.2 実験の方法	12
3.3 測定方法	13
3.4 実験データの処理作業	16
4. 実験結果	28
4.1 運転データ	28
4.2 実験データ	28
5. 考察	68
6. おわりに	73
参考文献	74
付録 1 ディジタルデータ収録装置の仕様	75
付録 2 制御棒位置データ変換プログラム	79
付録 3 測定チャネルデータ選択プログラム	86
付録 4 制御棒位置・反応度データ変換プログラム	93
付録 5 MATLAB 作図コマンドプログラムの一例	101

Contents

1. Introduction	1
2. VHTRC	2
2.1 Reactor Assembly	2
2.2 Operational Conditions	3
3. Experiment	12
3.1 Outline	12
3.2 Method of Experiment	12
3.3 Method of Measurement	13
3.4 Data Processing	16
4. Results of Experiment	28
4.1 Operational Data	28
4.2 Experimental Data	28
5. Discussions	68
6. Concluding Remarks	73
References	74
Appendix 1 Specification of Digital Data Recorder	75
Appendix 2 Control Rod Position Data Conversion Program	79
Appendix 3 Measurement Channel Data Select Program	86
Appendix 4 Control Rod Position • Reactivity Data Conversion Program	93
Appendix 5 An exsample of MATLAB Plotting Command Program	101

1. はじめに

原子炉の運転中に反応度を正確に把握することは、原子炉の安全運転を保証するうえからも重要な項目の一つである。しかし、この反応度を中性子束や温度のように検出器を用いて直接計測するということは不可能であり、通常は原子炉出力（中性子束）の時間的な変化から間接的に推定を行っている。これまでに反応度推定を行う各種の理論や方式が提案され、各種の反応度計が試作され使用されてきた¹⁾。反応度の推定精度を少しでも向上させることは、原子炉の安全運転上も好ましい。そこで筆者らは $H\infty$ 制御理論を応用した反応度推定方式を提案し、推定精度の改善をはかるべく、新しい推定方式についての評価試験を開始した。今回の実験はその第1ステップとして、測定精度の高い実験が行える VHTRC（高温ガス炉臨界実験装置）を対象に、評価試験を実施した。評価試験は作業内容から 2 つの Phase に分けて実施した。前半は反応度推定のために必要な中性子束信号の取得を行う Phase-1、後半は取得した中性子束信号をもとに $H\infty$ 制御理論による反応度推定を行う Phase-2 である。Phase-1 の実験は VHTRC の現場にて行い、Phase-2 の評価実験は、各種のパラメータによる繰り返し評価を行うために、取得したデータを研究室に持ち帰って実施した。本報告書では VHTRC において実施した Phase-1 の実験についてまとめたものであり、実験の方法、実験の回路や取得したデータを詳細に記載した。Phase-2 の反応度推定評価に関する詳細は別の報告書にまとめる予定である。また本報告書には、取得したデータを処理する汎用的なプログラムを作成したので付録に掲載した。このデータ処理プログラムは C 言語で作成されており、ワークステーションやパソコン上で使用可能であり、同様な形式のデータファイルを取り扱う場合には有効であると思われる。

2. V H T R C の概要²⁾

2.1 原子炉集合体

V H T R C は高温工学試験研究炉 (H T T R) の核的な安全性の検証を行うことを目的に、前身の半均質臨界実験装置 (S H E) を改造し、昭和 60 年 5 月 13 日に初回臨界を達成した原子力施設であり、東海研究所の開発試験室建屋に設置されている。

第 2.1 表に V H T R C の主な仕様項目を示す。

V H T R C の炉心は黒鉛ブロックを積層した 2 つの 1 / 2 集合体から構成される。各集合体は炉心黒鉛ブロックおよび反射体黒鉛ブロックから成り、垂直直径方向に約 2.4 m、水平軸方向に約 1.2 m 長の横置六角柱状の構造をしている。1 / 2 黒鉛集合体の一方が固定型(FIXED SIDE)になっており、もう一方の集合体が移動型(MOVABLE SIDE)となっている。V H T R C 運転停止中は 2 つの 1 / 2 黒鉛集合体を約 1.5 m 離れた位置に分離しておき、V H T R C を運転する時に 2 つの集合体を密着合体させ 1 つの炉心を構成する構造になっている。炉心黒鉛ブロックには燃料を装荷するための孔を備えた燃料用黒鉛ブロックと実験用制御棒を挿入できる制御棒黒鉛ブロックがあり、実験の目的に応じて燃料の装荷量や炉心配置などを比較的容易に変更できる。反射体黒鉛ブロックには多目的孔付き六角柱状のものと、台形柱状の 2 種類がある。燃料体は低濃縮ウランを用いた被覆粒子燃料コンパクトを黒鉛製のさやに装填した構造の燃料棒を使用している。

第 2.1 図に V H T R C の概念図を、第 2.2 図に黒鉛集合体の断面図を示す。

V H T R C の核計装設備は起動系 2 チャンネル、運転系 3 チャンネル、安全系 2 チャンネル、計 7 チャンネルで構成されている。各検出器は固定側 1 / 2 黒鉛集合体の側面に配置されている（第 2.2 図参照）。その他の計測信号としては炉心温度測定用の温度センサが黒鉛集合体の中に設置されている。

第 2.2 表に核計装の一覧表を示す。チャンネル 1, 2, 3, 7 には炉出力測定レンジ毎に炉出力高のアラームおよびスクラム信号を発信する回路が、チャンネル 5, 6 にはソースインターロックおよび炉周期短のアラーム、およびスクラム信号を、チャンネル 4 には炉周期短のアラーム、およびスクラム信号を発信する回路が設けられている。また、チャンネル 3, 7 には炉出力 1.2 W でスクラム信号を発信する回路が設けられている。

V H T R C の運転制御設備は制御棒ならびに安全棒装置および架台駆動機構から構成される。制御棒装置は固定側および移動側に各々 1 系統設置（一系統は中性子吸収棒 1 本で構成）されており、通常の運転制御時にはモーター駆動ボールナット方式で駆動されるが、スクラム時には圧縮空気によって急速炉心挿入される。制御棒駆動装置の全長は 2450 mm、駆動ストロークは 990 mm である。制御操作時の駆動最高速度は 240 mm/min である。

安全棒装置は固定側および移動側に各々 3 系統設置（一系統は中性子吸収棒 1 本で構成）されており、スクラム時には圧縮空気により、安全棒が高速で炉心に挿入される

(90%ストローク0.3sec以内)。安全棒の駆動ストロークは900mmである。

架台駆動機構は移動側1/2黒鉛集合体の架台を電動方式により移動し、固定側1/2黒鉛集合体と密着合体させる装置であり、移動ストロークは約1500mmである。

2.2 運転条件

今回の実験を実施した際のVHTRCの諸条件について述べる。VHTRCは燃料の装荷量や制御棒等の炉心配置を実験の目的に応じて比較的容易に変更できる特徴を備えていることを先にも述べた。今回の実験は燃料棒を280本装荷したVHTRC-1

(2)炉心構成のもとで行った。VHTRC-1(2)炉心の制御棒および燃料棒の装荷図を第2.3図に示す。

第2.3表にVHTRC-1(2)炉心構成データを、第2.4表に同炉心の核定数を示す。第2.5表にVHTRC-1(2)炉心の制御棒位置に対するロッドワースを求める多項式の係数を示す。

第2.4図にその係数を用いて計算した同炉心構成の制御棒反応度曲線を示す。

今回の実験は先ず0.1W付近の出力領域において8ケース(実験No.1～No.8)の実験を行い、次に1W付近に出力を上昇させて同じ8ケース(実験No.9～No.16)の実験を行った。各実験項目についての詳細は次章以降に記載した。

第2.1表 VHTRCの主要な仕様一覧表²⁾

項目	仕 様
最高熱出力	10W
最高集合体温度	210°C(昇温実験時)
降温時の集合体温度降下率	6°C/h以下(昇温実験時)
地震荷重	水平方向0.3Gと鉛直方向0.3Gの地震荷重が同時に作用
炉心形状	横置六角柱状ブロック積層構造
集合体寸法	六角柱径方向高さ:約2.4m、水平軸方向長:約2.4m
集合体重量	36t以下(スチールフレームを含む)
最高燃料装荷量	U-235 10.4kg
最高過剰反応度	0.5%△k/k
安全棒シャットダウン余裕	$k_{eff} < 0.975$
燃料	被覆粒子燃料コンパクト
燃料コンパクト	中空円筒形(内径:18mm、外径:36mm、高さ:36mm) ウラン含有量:約21gU/コンパクト 濃縮度:約2,4,6wt%
炉心黒鉛ブロック	構成:被覆粒子燃料用黒鉛ブロック、実験用制御棒黒鉛ブロック 寸法:平径300mm、長さ1200mm
反射体黒鉛ブロック	形状:六角柱状ブロックまたは台形柱状 寸法:平径300mm、長さ1200mm(六角柱状ブロック) 高さ150mm、長さ1200mm(台形状ブロック)
燃料貯蔵容量	U-235 35kg
制御装置 制御棒装置	2セット SUS被覆シリンダー形カドミウム ストローク:990mm
安全棒装置 中性子吸收棒	6セット SUS被覆シリンダー形カドミウム ストローク:900mm
反応度制御性 移動側1/2ブロック集合体 間隔反応度 反応度付加率	$>0.3\% \Delta k/k/cm$ $<0.015\% \Delta k/k/s$
安全棒 反応度 反応度付加率	$>3\% \Delta k/k/6\text{ rods}$ $<1.4\% \Delta k/k/1\text{ rod}$
制御棒 反応度 反応度付加率	$>1\% \Delta k/k/2\text{ rods}$ $<0.01\% \Delta k/k/s$
核計装備 中性子束 γ線	6チャンネル 1チャンネル
安全保護回路 スクラム条件 アラーム条件	14 6

第2.2表 VHTRCの核計装設備一覧表

チャンネルNo.	検出器	計装用途	安全動作
1	中性子電離箱 線形出力計	運転系	炉出力高アラーム、スクラム信号の発信
2	中性子電離箱 線形出力計	運転系	炉出力高アラーム、スクラム信号の発信
3	中性子電離箱 線形出力計	安全系	炉出力高アラーム、スクラム信号の発信 炉出力12Wでスクラム信号を発信
4	中性子電離箱 炉周期計付対数出力計	運転系	炉周期短アラーム、スクラム信号の発信
5	比例計数管 炉周期計付対数計数率計	起動系	ソースインターロック 炉周期短のアラーム、スクラム信号の発信
6	比例計数管 炉周期計付対数計数率計	起動系	ソースインターロック 炉周期短のアラーム、スクラム信号の発信
7	γ 線電離箱 線形出力計	安全系	炉出力高アラーム、スクラム信号の発信 炉出力12Wでスクラム信号を発信

第2.3表 VHTRC-1(2) 炉心構成データ

項目	挿入位置	
制御棒位置	移動側	V - 6
	固定側	E - 3
安全棒位置	S R - 1	C - 1
	S R - 2	Z - 4
	S R - 3	B - 6
	S R - 4	Y - 3
	S R - 5	F - 4
	S R - 6	W - 1

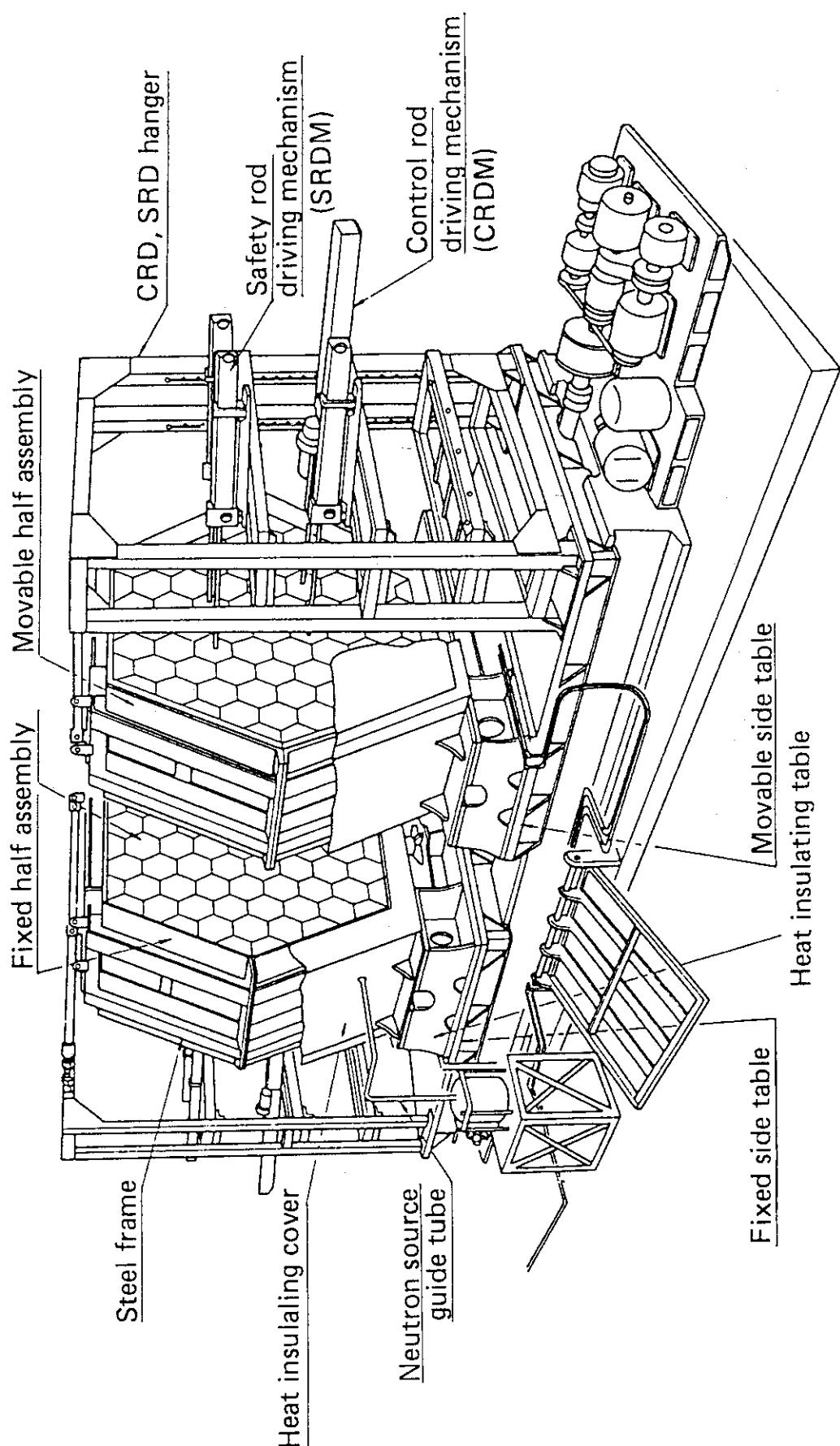
第2.4表 VHTRC-1(2) 炉心の核定数

i	$\beta_{eff}(i)$	$\lambda(i)$
1	0.000278075	0.0127
2	0.00151526	0.0317
3	0.00137974	0.116
4	0.00298627	0.311
5	0.000950742	1.40
6	0.000196589	3.87
$\Sigma \beta_i$	0.007306676	
Neutron Life time(msec)		1.23485

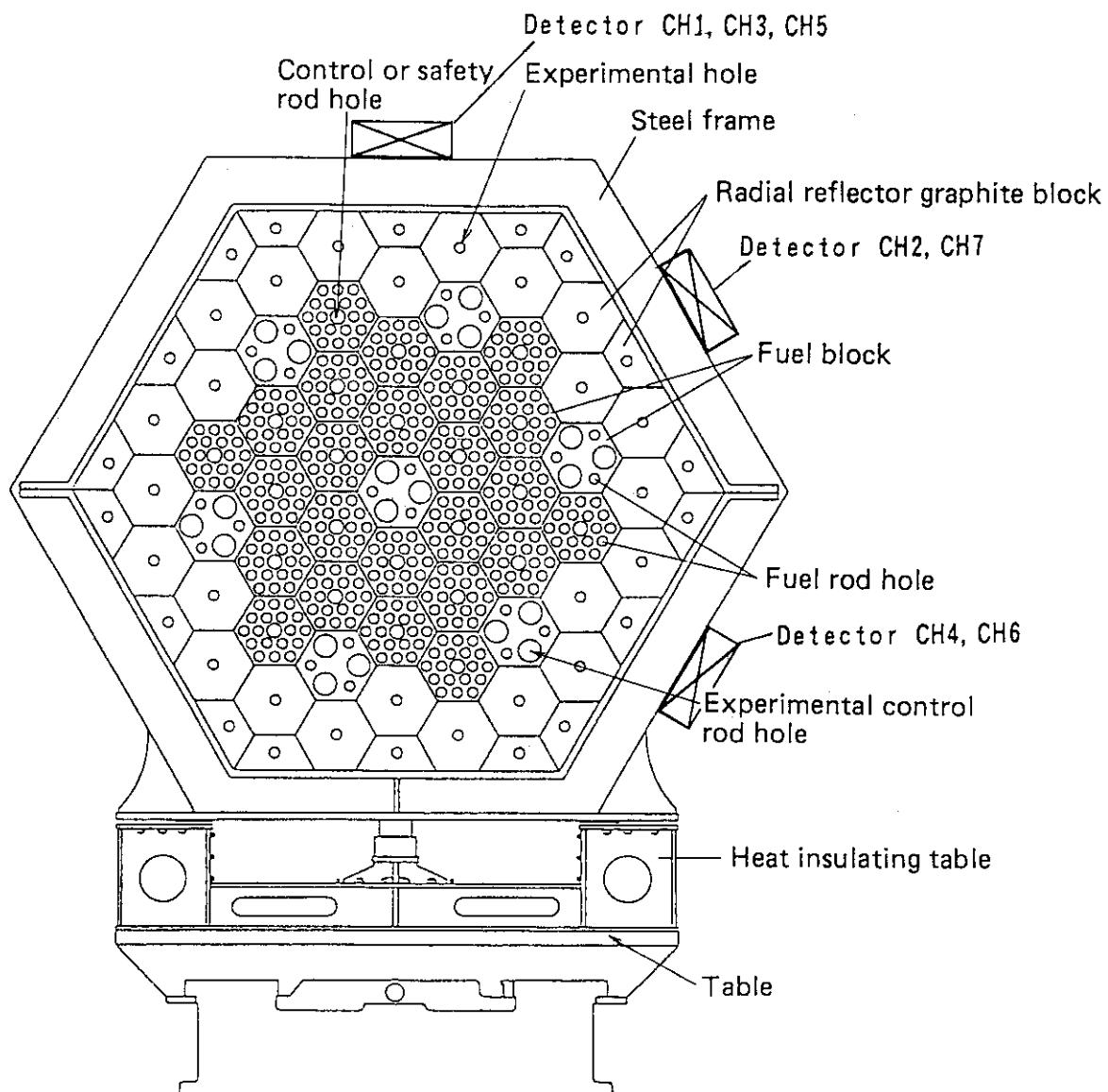
第2.5表 制御棒位置・ロッドワース多項式の係数

次数	移動側制御棒	固定側制御棒
0	0.11822E+03	0.12821E+03
1	-0.24733E+00	-0.29634E+00
2	0.80427E-04	0.15759E-03
3	0.11144E-06	0.65899E-07
4	-0.62794E-10	-0.55398E-10

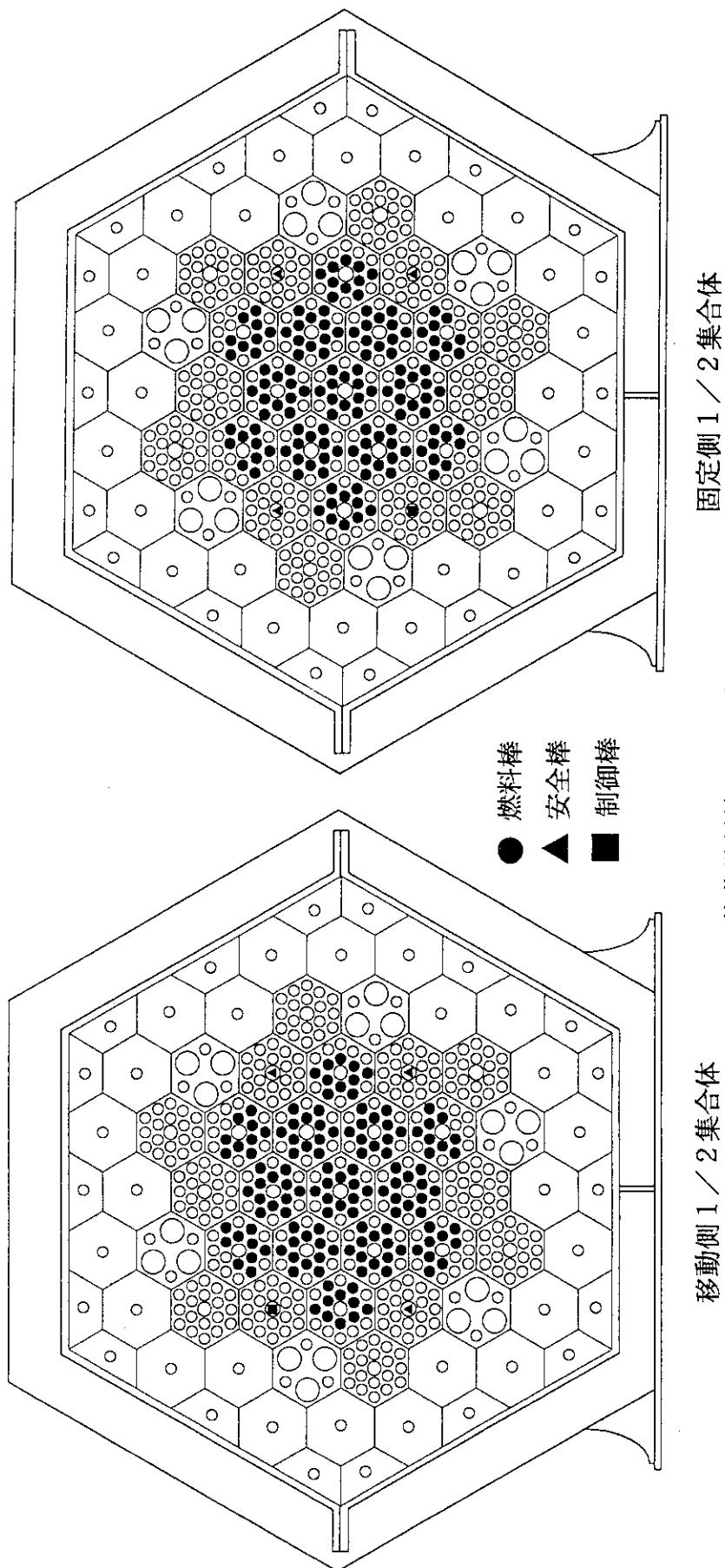
炉心構成：VHTRC-1(2)炉心



第2.1図 VHTRCの概念図

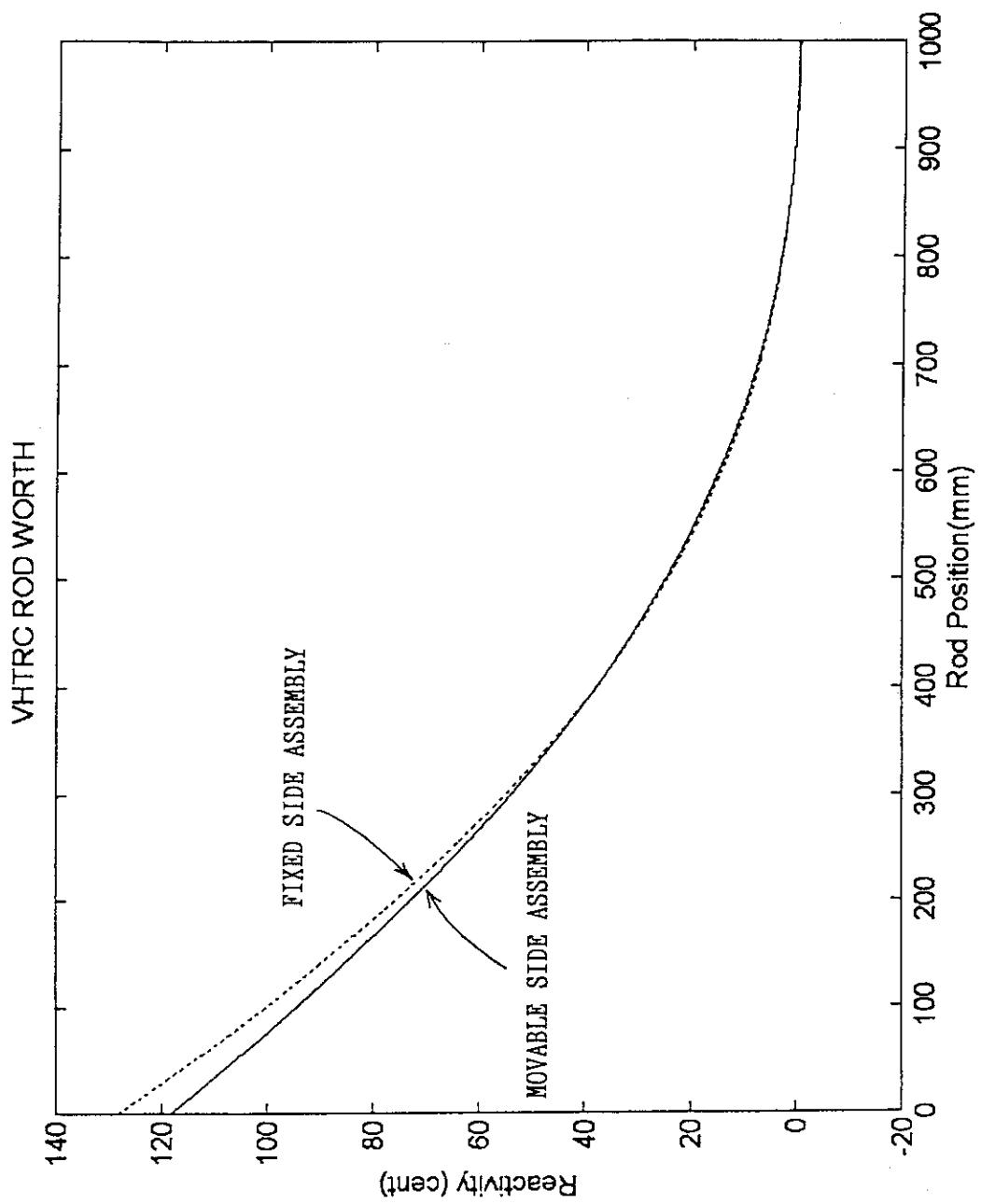


第2.2図 横置六角柱状黒鉛集合体の断面図



炉心 VHTRC-1 (2) 炉心
1995年9月21日

第2.3図 VHTRC-1(2) 炉心の燃料棒装荷図



第2.4図 VHTRC-1(2)炉心の制御棒反応度曲線

3. 実験

3.1 実験の概要

今回の実験は、VHTRC（炉心構成:VHTRC-1(2)以下同じ）において $H\infty$ 反応度推定器の評価試験を行うための予備実験として、VHTRCの制御棒位置をいろいろな操作パターンに基づいて変更した場合の炉出力信号の変化を観測するとともに、その信号をデータ収録装置に収録することを目的として実施した。データ収録装置には、操作した制御棒の位置信号と核計装設備から得られる中性子束信号および γ 線検出器信号を同時に収録するようにした。

第3.1表に実験項目の一覧表を示す。同表において実験No.1～No.6は制御棒位置をステップ状に変更した場合の炉出力の応答を調べるものであり、実験No.7およびNo.8は制御棒位置をそれぞれ台形波状、三角波状に変更した場合の炉出力の応答を調べるものである。No.9～No.16の実験はNo.1～No.8と全く同じ制御棒操作を炉出力を変えて行うものである。No.1～No.8の実験は0.1W付近の炉出力で行い、No.9～No.16の実験は1～5W付近の炉出力で行う。これは炉出力レベルを変えた場合の反応度推定結果に及ぼす影響の有無を評価することがその目的である。

今回の実験を実施するにあたっては、VHTRCの計装制御系に改造等の手を加えること無しに、出来る限り多くの計測信号を収録できるように計画した。

今回の実験はデータ収録までの作業をVHTRC建屋において実施し(Phase-1)、収録したデータを用いた $H\infty$ 反応度推定の評価試験は研究棟側に持ち帰って実施することとした(Phase-2)。これは、反応度推定の評価試験を、同じ収録データを用いていろいろなパラメータ条件のもとで、何回も繰り返し実施したいことと、そのデータ処理にある程度の時間を要することから2つのPhaseに分けて実施する事とした。

3.2 実験の方法

3.2.1 热出力

今回の実験はVHTRCの熱出力0.1～0.6W付近の低出力領域と、1～5W付近の高出力領域の2段階の領域で行った。

- | | |
|-------------|-----------|
| (1) VHTRC出力 | 0.1～0.6 W |
| (2) VHTRC出力 | 1～5 W |

3.2.2 制御棒操作

VHTRCが出力領域において臨界状態で安定に運転されている場合に、ステップ状、台形状および三角波状の反応度変化を、制御棒操作によって与えた。VHTRCの制御棒は固定側1/2黒鉛ブロック集合体に設置されているもの1系統と、移動側1/2黒鉛ブロック集合体に設置されているものが1系統あるが、実験中は固定側の制御棒を全引

き抜き状態にし、炉出力の変更操作はすべて移動側の制御棒によって行った。制御棒位置の変更操作はV H T R Cの制御盤にて、運転員の手動操作によって行った。

第3.1図に制御棒位置の変更操作のパターンを示す。

(1) 制御棒位置をステップ状に変化

ステップ状の反応度変化を与える実験は、制御棒を最高速度で操作することとするが、制御棒駆動系の設定できる最高速度が比較的ゆっくりしているため、実際は台形状に近い反応度変化を与えることになる。

第3.1図(1)にステップ状反応度の印加状況を示す。反応度を変化させる方向はプラス／マイナス両方向について実施し、その変化量はそれぞれ2、4、6¢の3段階とした。

(2) 制御棒位置を台形状に変化

第3.1図(2)に示すように制御棒位置を台形状に変化させた。制御棒の引き抜き・挿入速度は同一速度とし、段階的に±2、±4、±6¢の台形状の反応度変化を与えた。第3.1図(2)の時間軸の数値は、これらの反応度変化を得るために計画的な目安の値であり、制御棒速度によって調整するものとした。反応度の変化量を正確に設定することに重点をおき、時間軸の数値は目安と考えた。

(3) 制御棒位置を三角波状に変化

第3.1図(3)に示すように制御棒位置を三角波状に変化させた。制御棒の引き抜き・挿入速度は同一速度とし、段階的に±2、±4、±6¢の三角波状の反応度変化を与えた。第3.1図(3)の時間軸の数値は、台形状の制御棒操作を行った場合と同様、計画的な目安の値である。

3.3 測定方法

3.3.1 計測信号

今回の実験で計測したV H T R Cの信号はリニアN炉出力信号4系統、ログN炉出力信号1系統、制御棒位置信号2系統である。

第3.2表に計測したV H T R Cの信号の一覧表を示す。これらの信号はデジタル方式のデータ収録装置によって収録した。アナログの炉出力信号については、50 msec毎に各信号を同時サンプリング、A/D変換し収録した。A/D変換の分解能は符号も含めて12ビットである。リニアN炉出力信号4系統については、V H T R C側から各チャネルのレンジ信号も出力されているが、今回の実験においては測定中にレンジを切り換えるようなモードは含まれていないため、レンジ信号の収録は行わなかった。制御棒位置信号は固定側、移動側ともそれぞれ4桁のB C Dコードで検出されているため、両方あわせて32点のデジタル信号入力としてアナログ信号と同時にデータ収録装置に収録した。ただし、実験No.1～No.6の制御棒位置信号は信号レベルの条件が満足せず、データ収録装置に収録できなかったため、急遽V H T R C既設のデータロガーに収録した。実験No.7～No.18の制御棒位置信号については、信号レベルのマッチングをとり直接デジタルデータ収録装置に収録した。デジタルデータ収録装置に収録した

データは各RUN No.毎にデータファイル名を付け、フロッピイディスクにセーブした。

3.3.2 実験回路

VHTRCの計測信号はすべてVHTRC制御盤に接続されているが、多くの計測信号が既設のVHTRCデータロガーにも集中入力されている。今回の実験においてはこのデータロガーを運転しないため、このデータロガーに入力されている信号ラインの一部を本実験回路の方に切り換えてデータの収録を行った。

第3.2図に実験回路のブロック図を示す。デジタルデータ収録装置に収録した信号は、VHTRCデータロガー用CAMAC装置に入力している信号のコネクタを一時的に分離し、今回の実験用に用意した同一規格のコネクタ(DDK57-10500-7型)側につなぎ替えることによって分岐した。

第3.3図にデータ収録用端子板の概念図を示す。つなぎ替えたコネクタは次の2箇所である。

- (1) CAMACステーションNo.20入力コネクタ
 - リニアN信号 : 4系統(フローティング入力)
 - ログN信号 : 1系統
- (2) CAMACステーションNo.4入力コネクタ
 - 制御棒位置信号 : 2系統 BCD4桁(固定側、移動側)

CAMACステーションNo.20に入力されていた炉出力信号のレベルは、各測定レンジのフルスケールに対し0~5V及び0~4Vに増幅されたものであるため、そのままデータ収録装置に入力した。制御棒位置信号は固定側、移動側とも4桁のBCDコードであり、オープンコレクタ渡しのTTLレベルであったため、データ収録用端子板の入力部で5VのDC電源ソースを3.3kΩの抵抗器を介して各ラインに与えた。

3.3.3 使用機器

今回の実験に使用した主な機器を次に示す。

- (1) デジタルデータ収録装置(日置電機製8825型メモリハイコーダ)^①
 - 炉出力信号および制御棒位置信号を収録したデジタルデータ収録装置は汎用的で、多くの優れた性能を備えている可搬型収録装置であり、その詳細な仕様を付録1に示す。ここでは同装置の主な特徴、今回の実験で利用した機能の概要や設定した入力条件等を述べる。

デジタルデータ収録装置の主な特徴

- ① A/D変換分解能 : 12ビット
- ② 最高サンプリング速度 : 200 kSample/sec(メモリレコーダファンクション選択時)但しレコーダファンクション選択時は最高560~120μsec

- (使用CH数によって変る)
- ③ サンプリング・A/D変換タイミング : 全チャンネル同時
 - ④ 最高アナログ入力チャンネル数 : 16 CH (フローティング入力)
 - ⑤ アナログ測定レンジ : 5 mV/DIV~20 V/DIV (12レンジ)
 - ⑥ ロジック入力チャンネル数 : 8 CH (4点/CH)
 - ⑦ 最高データ収納容量 : 4 Mデータ
 - ⑧ 外部記憶装置(フロッピィディスク) : 3.5インチ
(PC9801用2HD, IBM-PC用2DD, MS-DOS対応)
 - ⑨ 表示機能(プラズマディスプレイ) : 電圧軸、時間軸方向への波形拡大・圧縮可能
 - ⑩ 測定機能 : メモリレコーダ、レコーダ、X-Yレコーダ、FFT
 - ⑪ 記録機能 : 感熱記録方式、264 mm×30 m感熱記録紙
 - ⑫ 時間軸 : 500 μsec/DIV~5 min/DIV (18レンジ)
 - ⑬ 最高記録長 : 20,000 DIV(メモリレコーダ1~2CH使用時)
10,000 DIV(メモリレコーダ3~4CH使用時)
5,000 DIV(メモリレコーダ5~8CH使用時)
2,000 DIV(メモリレコーダ9~16CH使用時)
 - ⑭ 記録表示フォーマット : シングル、デュアル、クワッド、オクト、ヘキサ、X-Yシングル、X-Yクワッド

設定した入力条件

- ① サンプリング周期 : 50 msec 全CH同時サンプリング
- ② 選択ファンクション : メモリレコーダ・ファンクション
- ③ アナログ使用チャンネル : CH1~CH5
- ④ ロジック入力チャンネル : CHA~CHH (4点/CH)
- ⑤ アナログ信号測定レンジ : 500 mV/DIV(CH1~CH5)
- ⑥ 記録表示フォーマット : デュアル(上側: CH1~CH3、下側: CH4, CH5)
- ⑦ 記録表示時間軸 : 5 sec/DIV
- ⑧ 記憶データファイル名 : VHTEST01.DAT~VHTEST16.DAT

(1) 接続コネクタ端子板

接続コネクタ端子板はVHTRCのCAMACステーションに入力されていた信号を、

本実験のデータ収録装置に入力しやすいように、新たに製作した中継用の端子板であり、CAMAC用コネクタ、ケーブル、ハーモニカ端子、BNCコネクタ等から構成される。リニアN信号はフローティング入力的方式で各チャンネルとも絶縁型BNCコネクタを使用して端子板シャーシ・グラウンドから絶縁した。

制御棒位置信号は固定側、移動側ともハーモニカ端子を使用しデータ収録装置のロジックプローブが接続しやすい様にした。CAMACステーションに入力されていた信号を切り換えるためのコネクタはDDK57-10500-7型（第1電子工業製）を使用した。

3.4 実験データの処理作業

VHTRCにおける測定データは3.5インチおよび5インチのフロッピイディスクに収録し、研究棟側に持ち帰ってからデータの処理作業を行った。3.5インチのフロッピイディスクはデータ収録装置で収録したデータであり、5インチのフロッピイディスクはVHTRCのデータロガーで計測した制御棒位置信号を収録したものである。5インチのフロッピイディスクは実験の開始時点では制御棒位置信号がうまくデータ収録装置側に入力されなかつたため、実験No.1～6までについて、急遽VHTRCのデータロガーに制御棒位置信号のみを収録したものである。なお、VHTRCのデータロガーによって制御棒位置信号を収録した際のサンプリング周期は200msecである。

次に測定データについて行った変換処理等について述べる。まず、デジタルデータ収録装置に収録したデータは圧縮されているため、直接、数値として読み出すことができないため、ASCIIコードに変換する処理をPC98シリーズのパソコンを使用して行った。この変換処理プログラム¹⁾はデータ収録装置のメーカーからサポートされたものであり、時間軸データを添付したり、特定チャンネル(1CH)のデータのみを選択変換処理する機能を備えている。今回は収録した全チャンネルのデータについてそのままASCIIコードに変換した。その際、時間軸データをサンプリングされたデータの先頭位置に添付した。したがって変換後のデータファイルは1チャンネル分増えた形式になった。データファイルのサンプリングした各データはフローティング(F)タイプの電圧値であり、その桁数は小数点以下5桁までのランダムな桁数である。各データ間はセパレータとして“，”記号で分離されているが、後のデータ解析のために“，”記号のセパレータをスペース記号に変換した。変換にはエディタ・プログラムMIFES for windowsを使用した。データ数が膨大であるため全ファイルの変換には約1日を要した。

第3.3表にVHTRCにおいて収録したファイルの一覧表を示す。同表のVHTRC炉出力欄()内の数値は運転中実際に変更された炉出力の範囲を示す。

次にデータ収録装置に収録した制御棒位置信号は固定側、移動側の順に“1”或は“0”的バイナリ・コード32ビットで表現されているため、収録した制御棒位置信号をバイナリ・コードからFタイプの数値に変換する処理を行った。変換にはSPARC2ワークステーションを使用した。変換した制御棒位置信号データは、固定側、移動側の順にサンプリングした計測データの最後の部分に追加した。なお、その際バイナリ・コード32ビットは削除した。

制御棒位置信号データ変換プログラムはSPARC2ワークステーションのCコンパイラを用いて作成した。制御棒位置信号データ変換プログラムの詳細を付録2に示す。なお、パソコンからSPARC2ワークステーションへのデータ・ファイルの転送にはFTPコマンドを用いてイーサネットを介して行った。

次にデータ・ファイルのサイズをコンパクトにし、データ処理を進めやすくするため、時間軸データとリニアN信号(CH1)だけを抜き出したデータ・ファイルを作成した。このデータ・ファイルが、続報で報告する予定の反応度推定評価試験に使用されるファイルとなる。この処理もSPARC2ワークステーションを用いて行った。任意のチャンネルデータを抽出するプログラムをC言語を用いて作成した。このプログラムは同じ様なフォーマットのデータ・ファイルを取り扱う場合に汎用的に使用できる有効な機能を備えている。チャンネルデータを抽出するプログラムの詳細を付録3に示した。

これまでデータ・ファイルについて行ってきた変換処理の関係を第3.4表の一覧表にまとめた。

次に制御棒位置信号の変化から反応度変化量を求めるプログラムを作成し、変換処理を行った。移動側制御棒位置信号の変化から反応度変化量を求める多項式の係数は、先に示した第2.5表の値を用いた。変換した反応度の数値をデータ・ファイルの最後の部分に追加挿入した。VHTRCデータロガーによって計測したデータは1列目の数値はサンプリングの回数を示すインデックスNo.であったため、これを実際のサンプリング時間軸の数値に変換した。2列目は移動側の制御棒位置、3列目が固定側の制御棒位置であり、4列目に反応度の数値を追加挿入した。データ収録装置によって収録した実験No.7～No.16の制御棒位置データについても同様に反応度の数値を求め、最後のデータ・チャンネルに追加挿入した。

第3.4図に反応度の数値を追加挿入した実験No.1～No.6の制御棒位置信号データ・ファイルのフォーマットを示す。

第3.5図にデータ収録装置で収録したデータ・ファイルに反応度の数値を追加挿入した実験No.7～No.16のデータ・ファイルのフォーマットを示す。

制御棒位置データを反応度の数値に変換するプログラムの詳細を付録4に示した。

第3.5表に制御棒位置データについて変換処理を行ったファイルの一覧表を示す。実験No.1～No.6までの制御棒位置データ・ファイルはVHTRCのデータロガーを利用して収録したデータであり、データ収録装置で収録したデータ群とはサンプリング・タイムを異にしている。

第3.1表 VHTRCにおけるH∞反応度推定器の評価実験項目一覧表

No.	実験項目	VHTRC運転条件	制御棒操作方向	操作量
1	ステップ状応答試験	炉出力 0.1W	引き抜き	+ 2 ¢
2	〃	〃	挿入	- 2 ¢
3	〃	〃	引き抜き	+ 4 ¢
4	〃	〃	挿入	- 4 ¢
5	〃	〃	引き抜き	+ 6 ¢
6	〃	〃	挿入	- 6 ¢
7	台形状応答試験	〃	引き抜き／挿入	± 2 ¢, ± 4 ¢, ± 6 ¢ 操作パターン4.2 (2)
8	三角波状応答試験	〃	引き抜き／挿入	± 2 ¢, ± 4 ¢, ± 6 ¢ 操作パターン4.2 (3)
9	ステップ状応答試験	炉出力 1～5W	引き抜き	+ 2 ¢
10	〃	〃	挿入	- 2 ¢
11	〃	〃	引き抜き	+ 4 ¢
12	〃	〃	挿入	- 4 ¢
13	〃	〃	引き抜き	+ 6 ¢
14	〃	〃	挿入	- 6 ¢
15	台形状応答試験	〃	引き抜き／挿入	± 2 ¢, ± 4 ¢, ± 6 ¢ 操作パターン4.2 (2)
16	三角波状応答試験	〃	引き抜き／挿入	± 2 ¢, ± 4 ¢, ± 6 ¢ 操作パターン4.2 (3)

第3.2表 VHTRCにおけるH ∞ 反応度推定法の評価実験計測信号一覧表

No.	計測信号	TAG No.	測定レンジ	信号レベル	備 考
1	リニアN出力信号	CH. 1	$10^{-11} \sim 10^{-5}$ A	0~5V DC	CAMAC STATION 20の入力部で切換える
2	リニアN出力信号	CH. 2	"	0~5V DC	"
3	リニアN出力信号	CH. 3	"	0~5V DC	"
4	リニアN出力信号	CH. 7	"	0~5V DC	"
5	ログN出力信号	CH. 4	$10^{-11} \sim 10^{-4}$ A	0~4V DC	"
6	固定側制御棒位置信号	FIX	BCD 4桁	TTL	CAMAC STATION 4 の入力部で切換える
7	移動側制御棒位置信号	MOVE	BCD 4桁	TTL	"

第3.3表 H∞反応度推定法評価実験収録データファイル一覧表

No.	制御棒操作パターン	VHTRC炉出力	データファイル名	データ収録時間	データ数/CH
1		公称 0.1W (0.1~0.2)	VHTEST1.DAT	340 sec	6,801
2		公称 0.1W (0.3~0.12)	VHTEST2.DAT	520 sec	10,401
3		公称 0.1W (0.1~0.3)	VHTEST3.DAT	270 sec	5,401
4		公称 0.1W (0.3~0.1)	VHTEST4.DAT	270 sec	5,401
5		公称 0.1W (0.1~0.5)	VHTEST5.DAT	260 sec	5,201
6		公称 0.1W (0.5~0.1)	VHTEST6.DAT	360 sec	7,201
7		公称 0.1W (0.3~0.54)	VHTEST7.DAT	640 sec	12,801
8		公称 0.1W (0.32~0.44)	VHTEST8.DAT	400 sec	8,001
9		公称 1W (1~2.7)	VHTEST9.DAT	410 sec	8,201
10		公称 1W (2.6~1)	VHTEST10.DAT	410 sec	8,201
11		公称 1W (1~5)	VHTEST11.DAT	420 sec	8,401
12		公称 1W (4.7~1.2)	VHTEST12.DAT	410 sec	8,201
13		公称 1W (0.72~5)	VHTEST13.DAT	320 sec	6,401
14		公称 1W (4.9~1.2)	VHTEST14.DAT	320 sec	6,401
15		公称 1W (1.8~3)	VHTEST15.DAT	540 sec	10,801
16		公称 1W (2.1~2.93)	VHTEST16.DAT	510 sec	10,201

データ・サンプリング・タイム=50ms

第3.4表 データファイル変換処理一覧表

No.	制御棒操作パターン	オリジナルデータファイル	M S - D O S フォーマットファイル	セパレータコード変換ファイル	制御棒位置コード変換ファイル	CH1, CH2 *1) データ選択ファイル
1		VHTEST1.DAT	VHTM1.DAT	VHTMS1.DAT	VHTMSC1.DAT	VHTRC01.DAT
2		VHTEST2.DAT	VHTM2.DAT	VHTMS2.DAT	VHTMSC2.DAT	VHTRC02.DAT
3		VHTEST3.DAT	VHTM3.DAT	VHTMS3.DAT	VHTMSC3.DAT	VHTRC03.DAT
4		VHTEST4.DAT	VHTM4.DAT	VHTMS4.DAT	VHTMSC4.DAT	VHTRC04.DAT
5		VHTEST5.DAT	VHTM5.DAT	VHTMS5.DAT	VHTMSC5.DAT	VHTRC05.DAT
6		VHTEST6.DAT	VHTM6.DAT	VHTMS6.DAT	VHTMSC6.DAT	VHTRC06.DAT
7		VHTEST7.DAT	VHTM7.DAT	VHTMS7.DAT	VHTMSC7.DAT	VHTRC07.DAT
8		VHTEST8.DAT	VHTM8.DAT	VHTMS8.DAT	VHTMSC8.DAT	VHTRC08.DAT
9		VHTEST9.DAT	VHTM9.DAT	VHTMS9.DAT	VHTMSC9.DAT	VHTRC09.DAT
10		VHTEST10.DAT	VHTM10.DAT	VHTMS10.DAT	VHTMSC10.DAT	VHTRC10.DAT
11		VHTEST11.DAT	VHTM11.DAT	VHTMS11.DAT	VHTMSC11.DAT	VHTRC11.DAT
12		VHTEST12.DAT	VHTM12.DAT	VHTMS12.DAT	VHTMSC12.DAT	VHTRC12.DAT
13		VHTEST13.DAT	VHTM13.DAT	VHTMS13.DAT	VHTMSC13.DAT	VHTRC13.DAT
14		VHTEST14.DAT	VHTM14.DAT	VHTMS14.DAT	VHTMSC14.DAT	VHTRC14.DAT
15		VHTEST15.DAT	VHTM15.DAT	VHTMS15.DAT	VHTMSC15.DAT	VHTRC15.DAT
16		VHTEST16.DAT	VHTM16.DAT	VHTMS16.DAT	VHTMSC16.DAT	VHTRC16.DAT

*1) CH1: 時間軸データ CH2: 核計装 CH1 データ

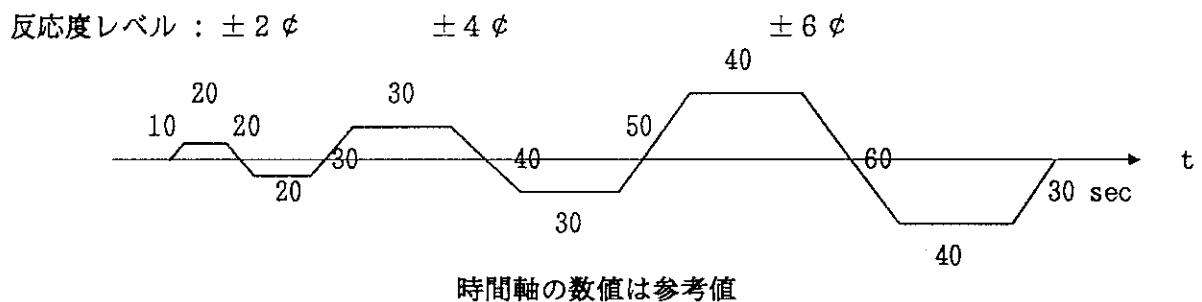
第3.5表 制御棒データファイル変換処理一覧表

No.	制御棒操作パターン	オリジナルデータファイル	反応度(Φ)変換ファイル	制御棒位置データ数/CH	最終データファイルのチャンネル数	データサンプリングタイム
1		VHTROD1.DAT	VHCENT01.DAT	1,778	4	0.20 sec
2		VHTROD2.DAT	VHCENT02.DAT	2,619	"	"
3		VHTROD3.DAT	VHCENT03.DAT	1,481	"	"
4		VHTROD4.DAT	VHCENT04.DAT	1,691	"	"
5		VHTROD5.DAT	VHCENT05.DAT	1,323	"	"
6		VHTROD6.DAT	VHCENT06.DAT	1,772	"	"
7	$\pm 2\Phi \pm 4\Phi \pm 6\Phi$ 	VHTMSC7.DAT	VHCENT07.DAT	12,801	9	0.05 sec
8	$\pm 2\Phi \pm 4\Phi \pm 6\Phi$ 	VHTMSC8.DAT	VHCENT08.DAT	8,001	"	"
9		VHTMSC9.DAT	VHCENT09.DAT	8,201	"	"
10		VHTMSC10.DAT	VHCENT10.DAT	8,201	"	"
11		VHTMSC11.DAT	VHCENT11.DAT	8,401	"	"
12		VHTMSC12.DAT	VHCENT12.DAT	8,201	"	"
13		VHTMSC13.DAT	VHCENT13.DAT	6,401	"	"
14		VHTMSC14.DAT	VHCENT14.DAT	6,401	"	"
15	$\pm 2\Phi \pm 4\Phi \pm 6\Phi$ 	VHTMSC15.DAT	VHCENT15.DAT	10,801	"	"
16	$\pm 2\Phi \pm 4\Phi \pm 6\Phi$ 	VHTMSC16.DAT	VHCENT16.DAT	10,201	"	"

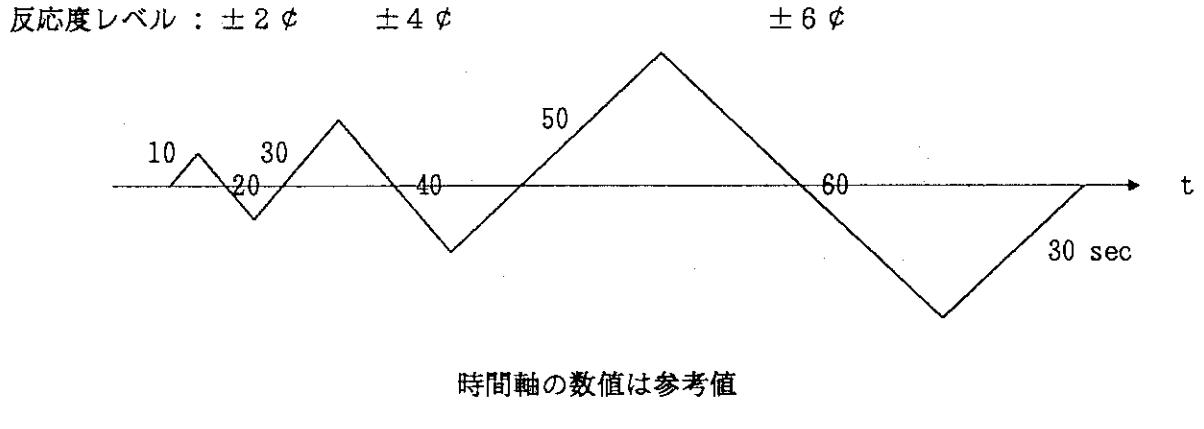


反応度レベル : $\pm 2 \phi$, $\pm 4 \phi$, $\pm 6 \phi$
制御棒操作速度 : 可能な最大速度

(1) 制御棒位置をステップ状に変化

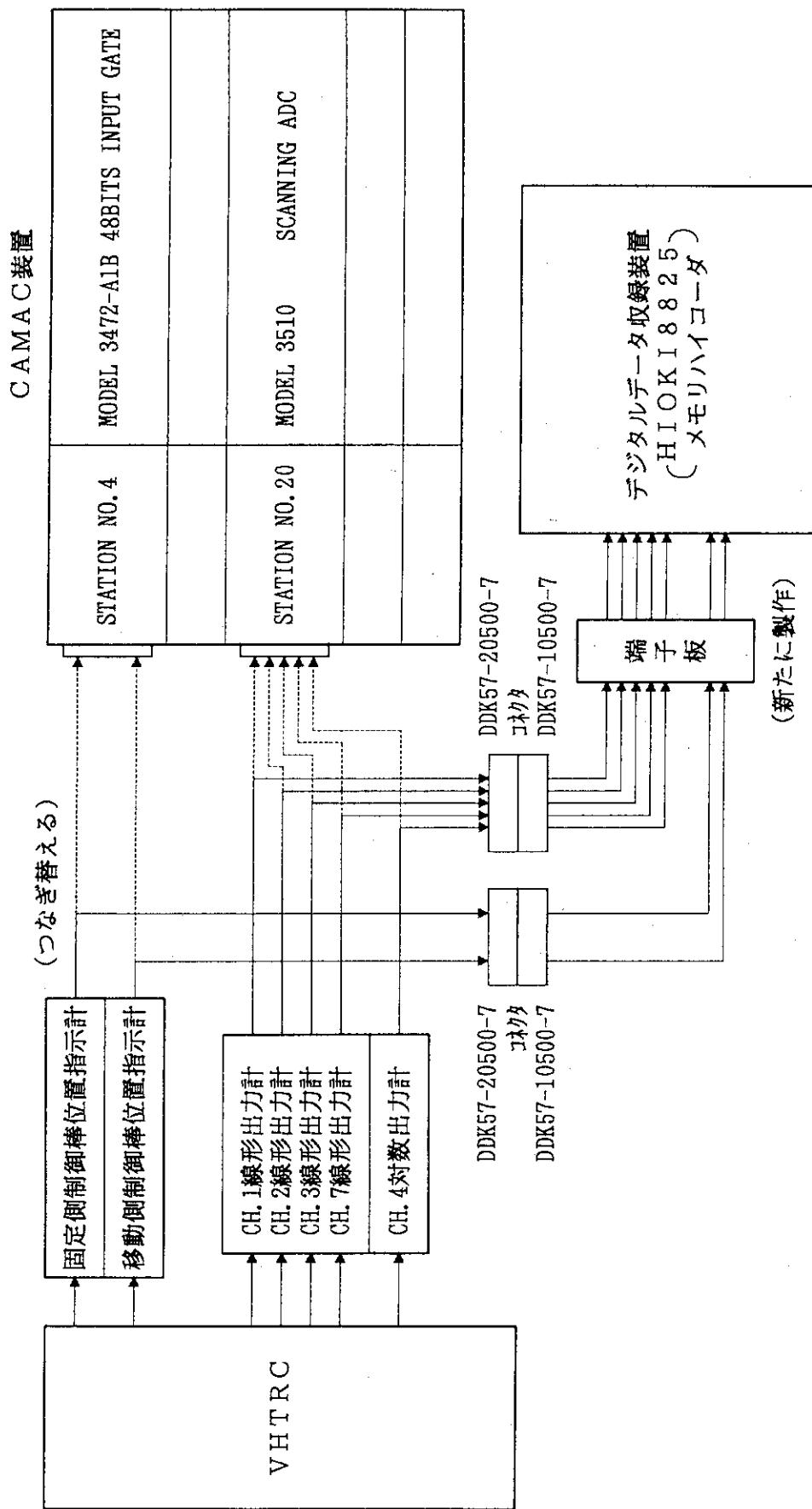


(2) 制御棒位置を台形状に変化

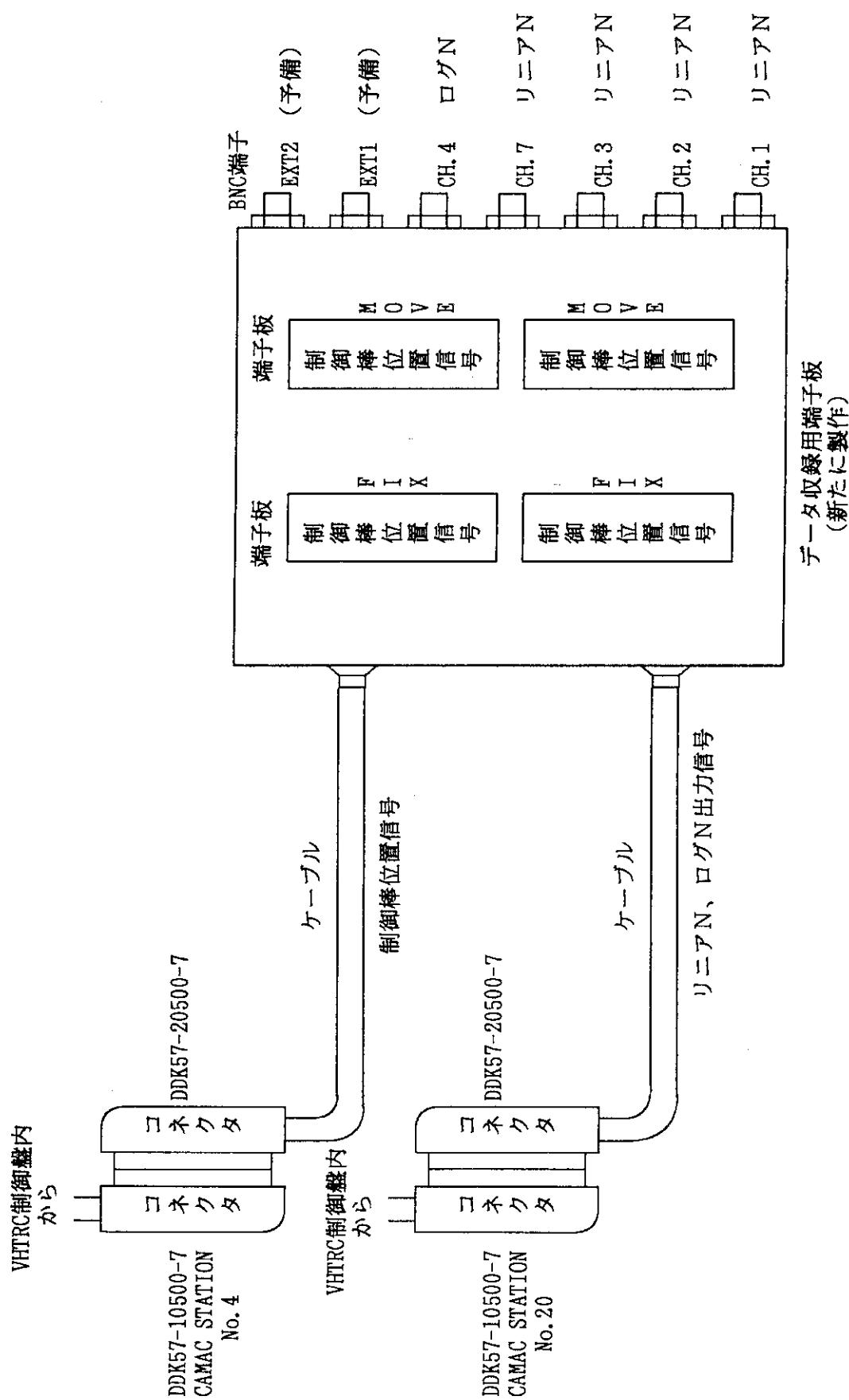


(3) 制御棒位置を三角波状に変化

第3.1図 制御棒位置の変更操作パターン



第3.2図 VHTRCにおけるH∞反応度推定法評価実験回路ブロック図

第3.3図 データ収録用端子板
(新たに製作)

時間軸 データ	移動側 制御棒位置 データ	固定側 制御棒位置 データ	反応度 データ
10.5F	10.5F	10.5F	10.5F
"	"	"	"
"	"	"	"
"	"	"	"

第3.4図 制御棒位置データ・ファイルのフォーマット

時間軸 データ	CH 1 データ	CH 2 データ	CH 3 データ	CH 7 データ	CH 4 データ	固定側 制御棒位置 データ	移動側 制御棒位置 データ	反応度 データ
10.5F	10.5F	10.5F	10.5F	10.5F	10.5F	10.5F	10.5F	10.5F
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"

第3.5図 VHTRC実験データ・ファイルのフォーマット

4. 実験結果

4.1 運転データ

今回の実験は1995年9月21日および22日の2日間にわたって実施した。21日には実験No.1から8までを0.1～0.6W付近の炉出力領域で行い、22日には実験No.9から16までを、炉出力1～5W付近で行った。

第4.1図にVHTRC制御盤の炉出力記録計で記録した21日の炉出力運転データを、第4.2図に22日の炉出力運転データを示す。いずれも核計装CH1の線形出力信号を記録したものである。

4.2 実験データ

第4.3図～第4.10図に、炉出力0.1W付近の領域における実験データを、第4.11図～第4.18図に、炉出力1W～5W付近の領域における実験データを示す。縦軸はデータ収録装置で入力した信号の電圧(V)を単位とし、横軸はデータ収録の経過時間(sec)を示している。実験データのグラフ化はすべて、PC98シリーズのパソコン上でWindows対応のMATLABプログラムを用いて行った。第4.3図(実験No.1)～第4.8図(実験No.6)には制御棒位置信号は含まれていない。実験No.1～16の制御棒位置信号のみを第4.19図～第4.34図に示した。制御棒位置信号はデータファイル内ではmm単位の数値で表現されているが、第4.9図～第4.18図内の制御棒位置信号については、グラフ化する際にMATLABプログラムによって便宜上電圧値に変換して、同一グラフにプロットした。そのスケール変換は制御棒位置0～1000mmを0～5Vに対応させている。

付録5にMATLABプログラムで実験データをグラフ化するコマンドファイルの一例を示す。

第4.1表にグラフプロット・コマンドファイルの一覧表を示す。

第4.2表に入力各チャネルのプロセス量と入力電圧との関係を示した。

第4.35図～第4.50図に制御棒位置を変更したトランジエント部分を時間軸方向に拡大したグラフを示す。図中には収録したデジタル値から直接読み取った制御棒位置の値とその時間を示してある。それらの値から求めた制御棒の駆動速度を第4.3表に示す。制御棒位置を台形状あるいは三角波状に変化させた場合の制御棒駆動速度は、操作方向を切り換える度に設定し直しているため、多少のばらつきがある。これは制御棒速度設定器の設定値が同じであっても、制御棒を引き抜く場合と挿入する場合とでは、駆動機構の特性によって、その駆動速度に差が生じるため、駆動方向を切り換える度に設定値を変更し、できるだけ両方向の操作に対し同一速度になるようにしたためである(引き抜き速度に対し、挿入速度の方が遅い)。

第4.51図～第4.66図に制御棒位置信号データから反応度変化量を求めた結果を示す。

第4.1表 データプロット・コマンドファイル一覧表

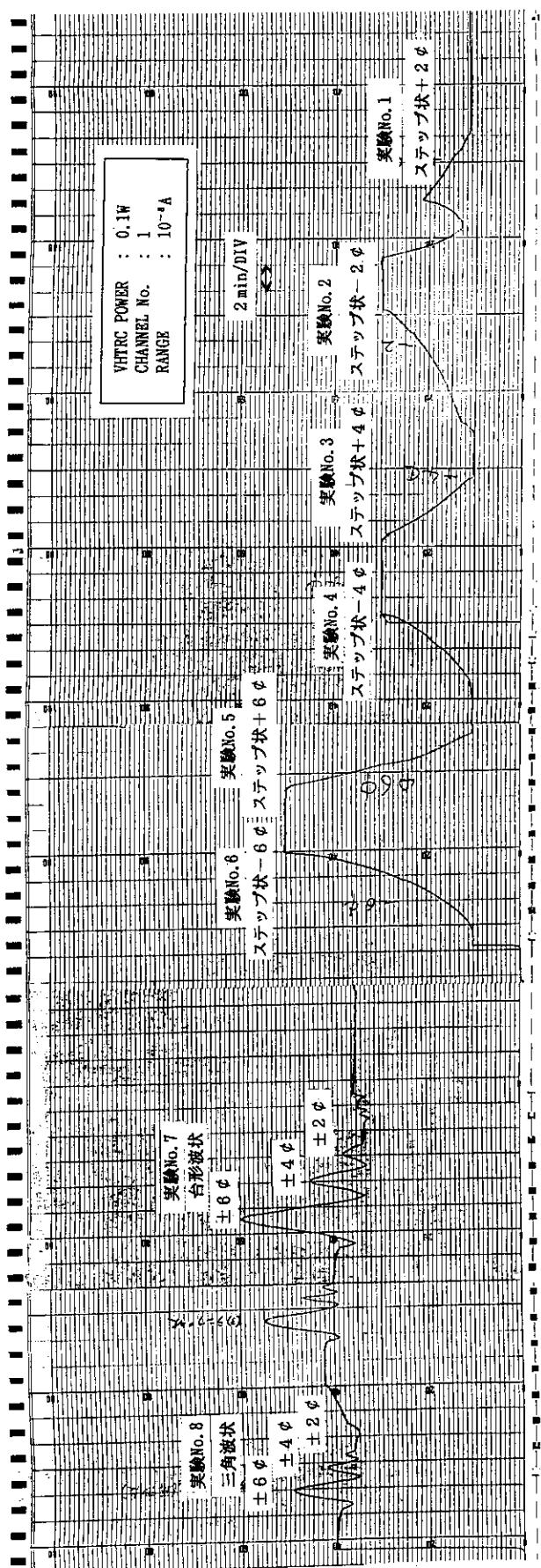
No.	制御棒操作パターン	プロット対象 ファイル	プロット コマンド ファイル	プロット データ数/CH	プロット データ チャンネル数
1		VHTMSC1.DAT	VHTEST01.M	6,801	8
2		VHTMSC2.DAT	VHTEST02.M	10,401	"
3		VHTMSC3.DAT	VHTEST03.M	5,401	"
4		VHTMSC4.DAT	VHTEST04.M	5,401	"
5		VHTMSC5.DAT	VHTEST05.M	5,201	"
6		VHTMSC6.DAT	VHTEST06.M	7,201	"
7		VHTMSC7.DAT	VHTEST07.M	12,801	"
8		VHTMSC8.DAT	VHTEST08.M	8,001	"
9		VHTMSC9.DAT	VHTEST09.M	8,201	"
10		VHTMSC10.DAT	VHTEST10.M	8,201	"
11		VHTMSC11.DAT	VHTEST11.M	8,401	"
12		VHTMSC12.DAT	VHTEST12.M	8,201	"
13		VHTMSC13.DAT	VHTEST13.M	6,401	"
14		VHTMSC14.DAT	VHTEST14.M	6,401	"
15		VHTMSC15.DAT	VHTEST15.M	10,801	"
16		VHTMSC16.DAT	VHTEST16.M	10,201	"

第4.2表 各チャンネルの入力電圧とプロセス量との関係

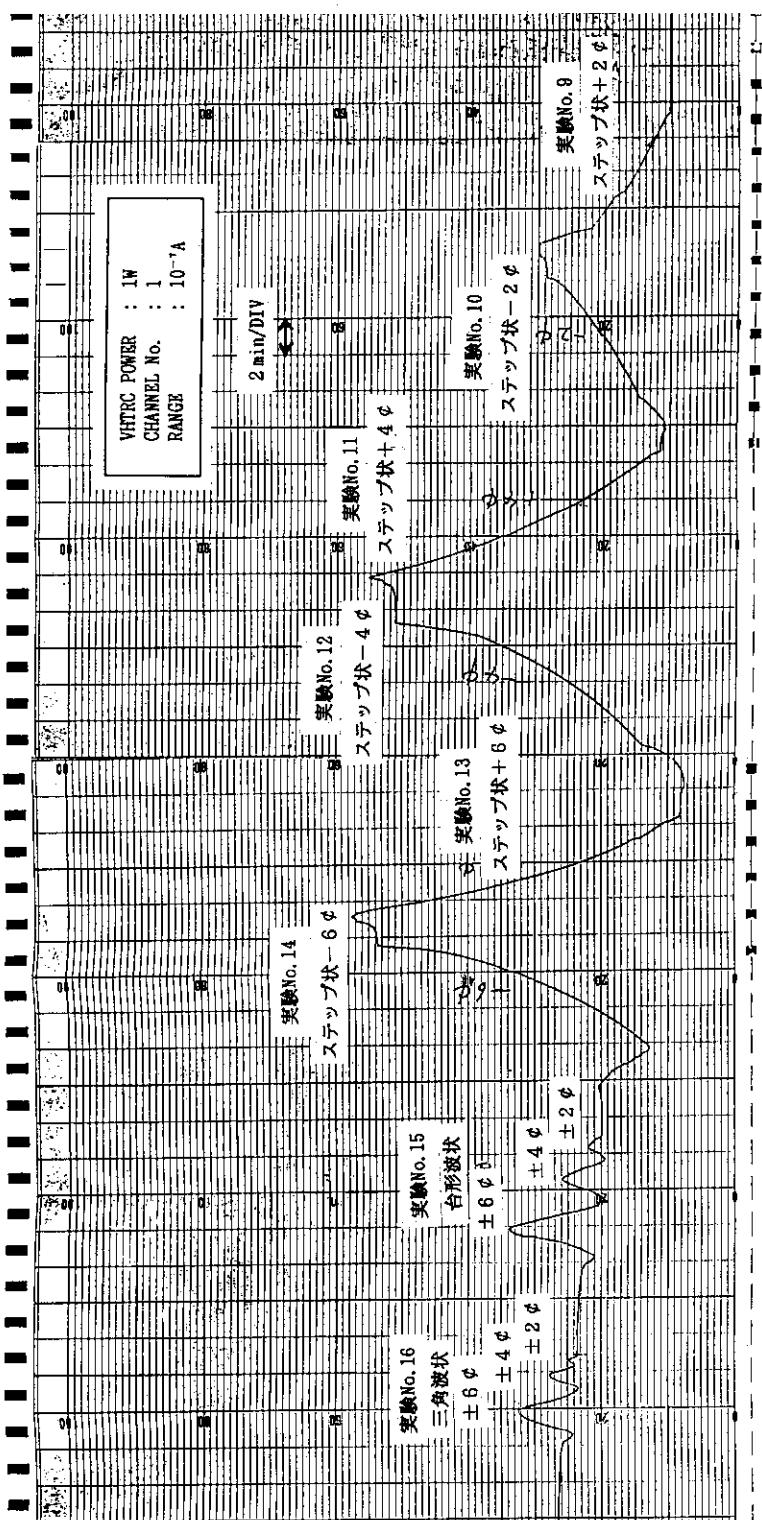
チャンネルNo	項目	入力電圧レベル	プロセス量
1	CH1リニアN信号	5VDC	1×10^{-8} A (実験No1~8) 1×10^{-7} A (実験No9~16)
2	CH2リニアN信号	5VDC	1×10^{-8} A (実験No1) 1×10^{-8} A (実験No2~8) 1×10^{-7} A (実験No9~16)
3	CH3リニアN信号	5VDC	1×10^{-7} A
4	CH7リニアN信号	5VDC	1×10^{-7} A
5	CH4ログN信号	4VDC	1×10^{-7} A
6	固定側制御棒位置信号	5VDC	1,000mm
7	移動側制御棒位置信号	5VDC	1,000mm

第4.3表 移動側制御棒駆動速度一覧表

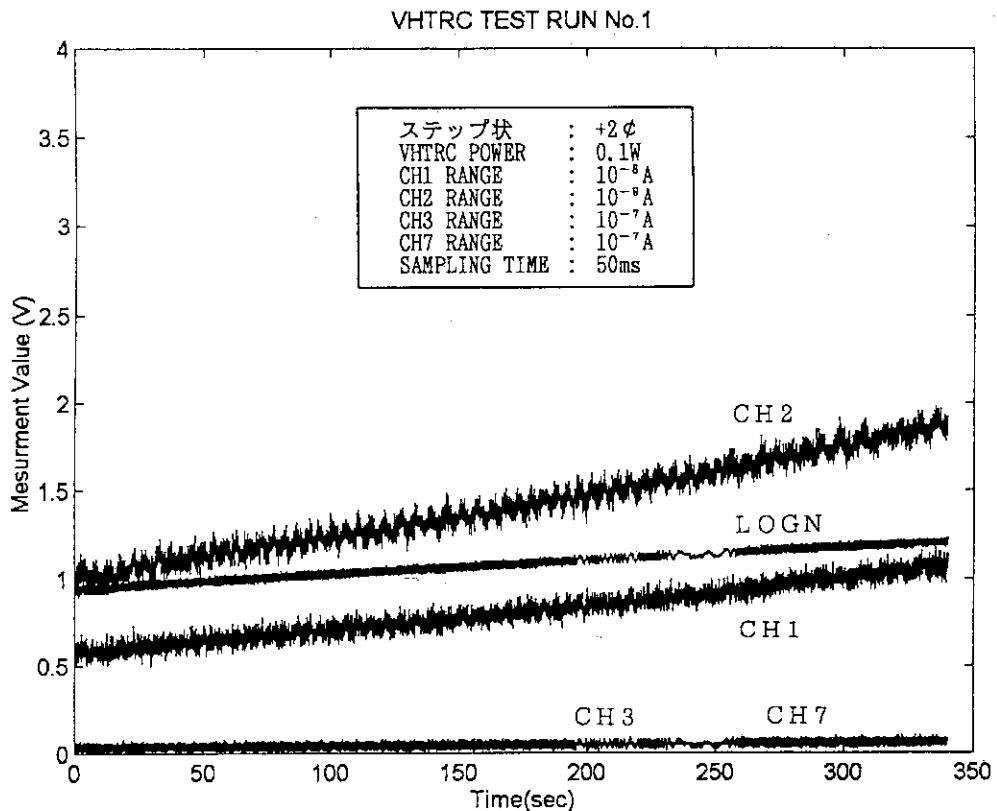
実験No	制御棒駆動速度	駆動方向	操作量	備考
1	4.06 mm/sec	引き抜き	ステップ状+2φ	炉出力: 0.1W
2	3.80	挿入	〃 -2	〃
3	4.04	引き抜き	〃 +4	〃
4	3.82	挿入	〃 -4	〃
5	4.03	引き抜き	〃 +6	〃
6	3.81	挿入	〃 -6	〃
7	1.77 1.36 1.69 1.22 1.67 1.17 2.47	引き抜き 挿入 引き抜き 挿入 引き抜き 挿入 引き抜き	台形状 +2 -4 +6 -8 +10 -12 +6	〃
8	1.79 1.56 1.69 1.38 1.58 1.27 1.41	引き抜き 挿入 引き抜き 挿入 引き抜き 挿入 引き抜き	三角波状 +2 -4 +6 -8 +10 -12 +6	〃
9	4.00	引き抜き	ステップ状+2φ	炉出力: 1~5W
10	3.77	挿入	〃 -2	〃
11	4.00	引き抜き	〃 +4	〃
12	3.78	挿入	〃 -4	〃
13	4.00	引き抜き	〃 +6	〃
14	3.77	挿入	〃 -6	〃
15	1.70 1.73 1.77 1.58 1.65 1.58 1.60	引き抜き 挿入 引き抜き 挿入 引き抜き 挿入 引き抜き	台形状 +2 -4 +6 -8 +10 -12 +6	〃
16	1.73 1.63 1.66 1.58 1.50 1.40 1.54	引き抜き 挿入 引き抜き 挿入 引き抜き 挿入 引き抜き	三角波状 +2 -4 +6 -8 +10 -12 +6	〃



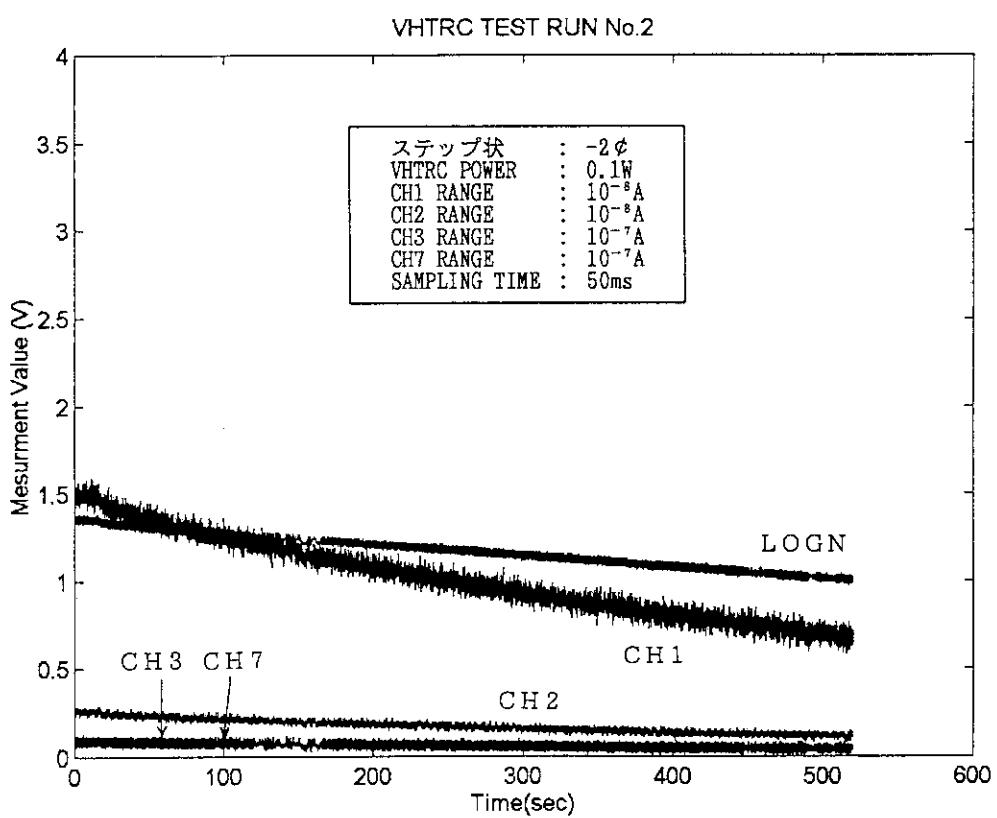
第4.1図 VHTRC線形出力計の記録（9月21日）



第4.2図 VHTRC線形出力計の記録（9月22日）

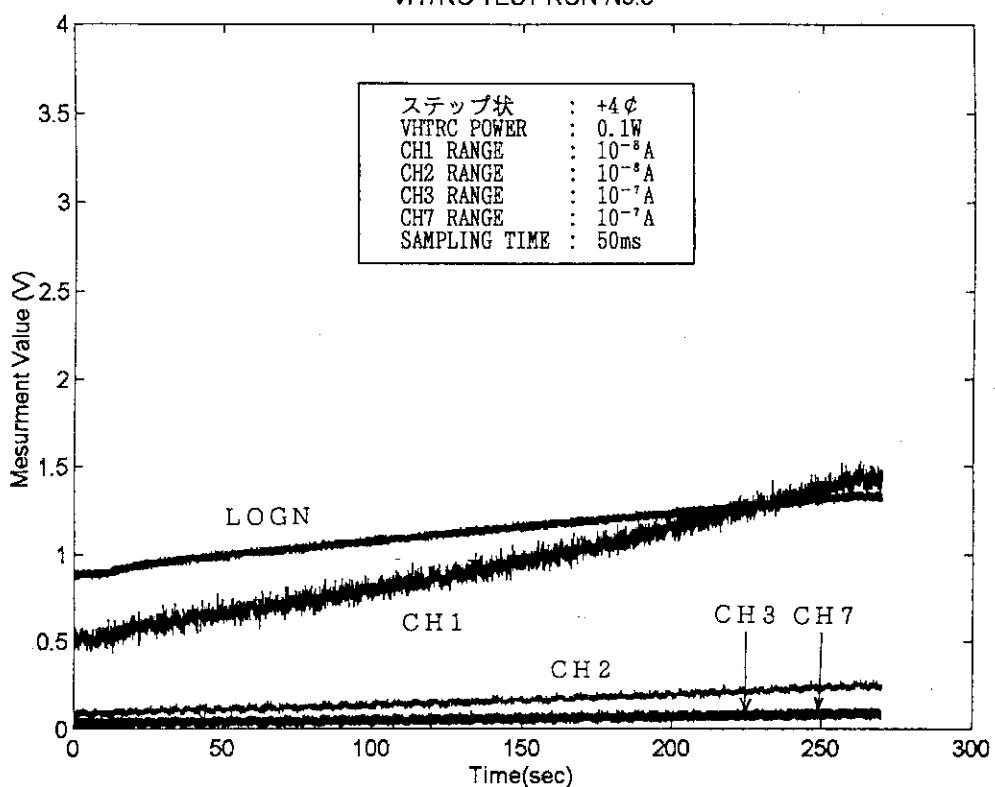


第4.3図 VHTRC実験データ（実験No.1）



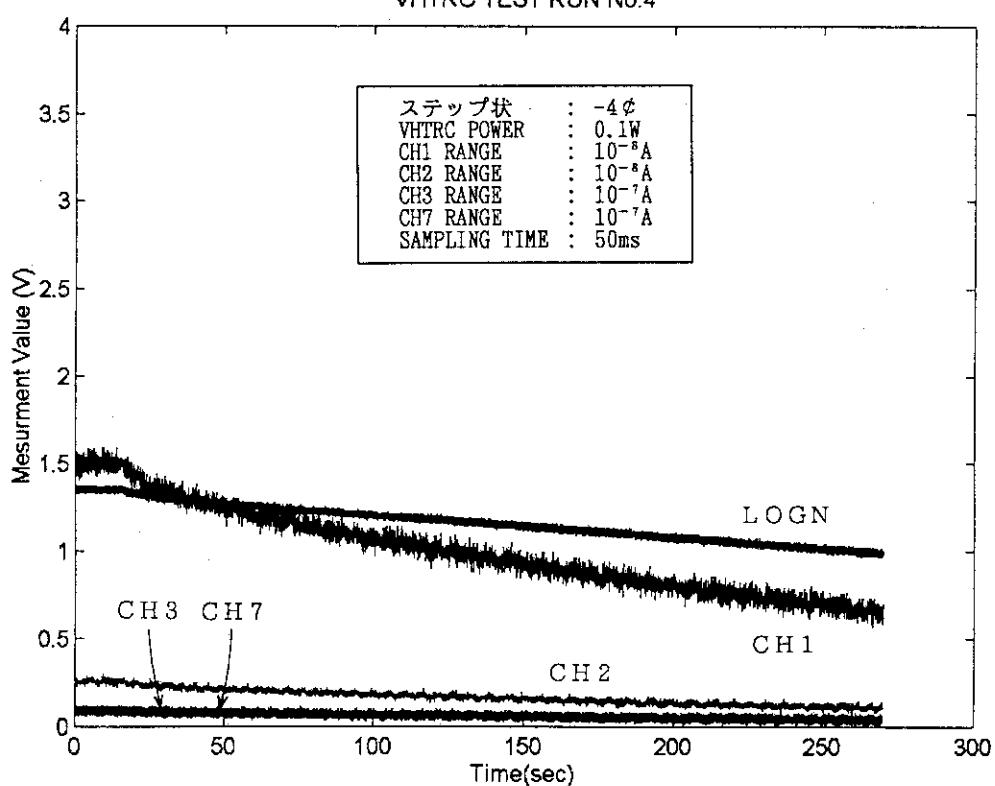
第4.4図 VHTRC実験データ（実験No.2）

VHTRC TEST RUN No.3



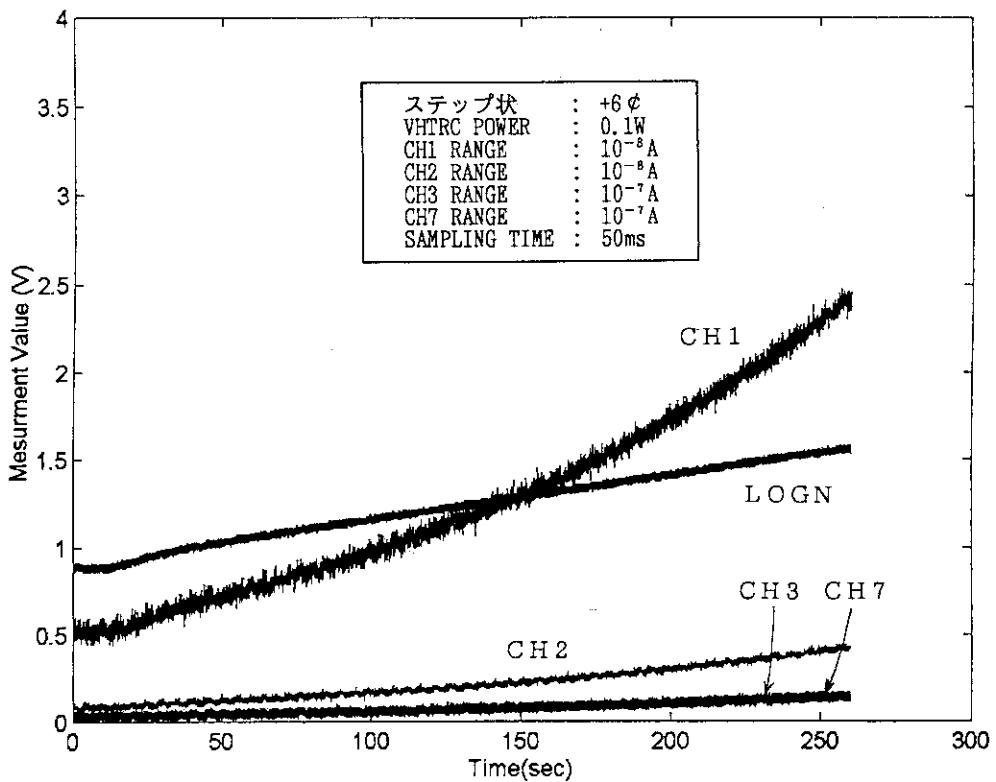
第4.5図 VHTRC実験データ（実験No.3）

VHTRC TEST RUN No.4



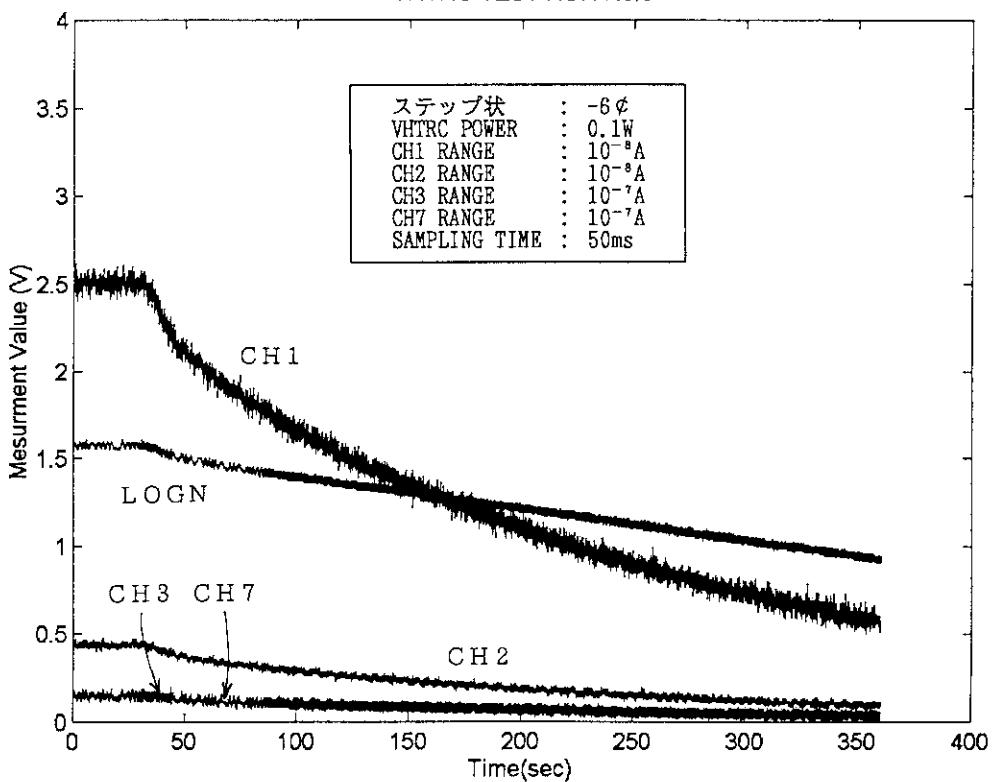
第4.6図 VHTRC実験データ（実験No.4）

VHTRC TEST RUN No.5



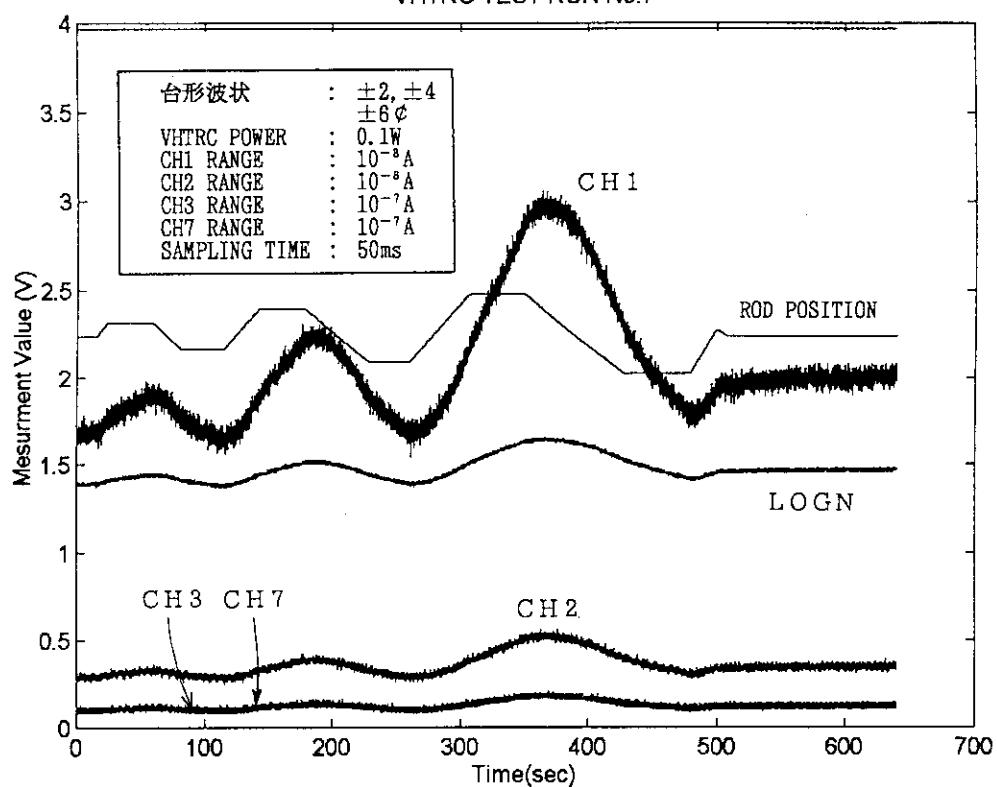
第4.7図 VHTRC実験データ（実験No.5）

VHTRC TEST RUN No.6



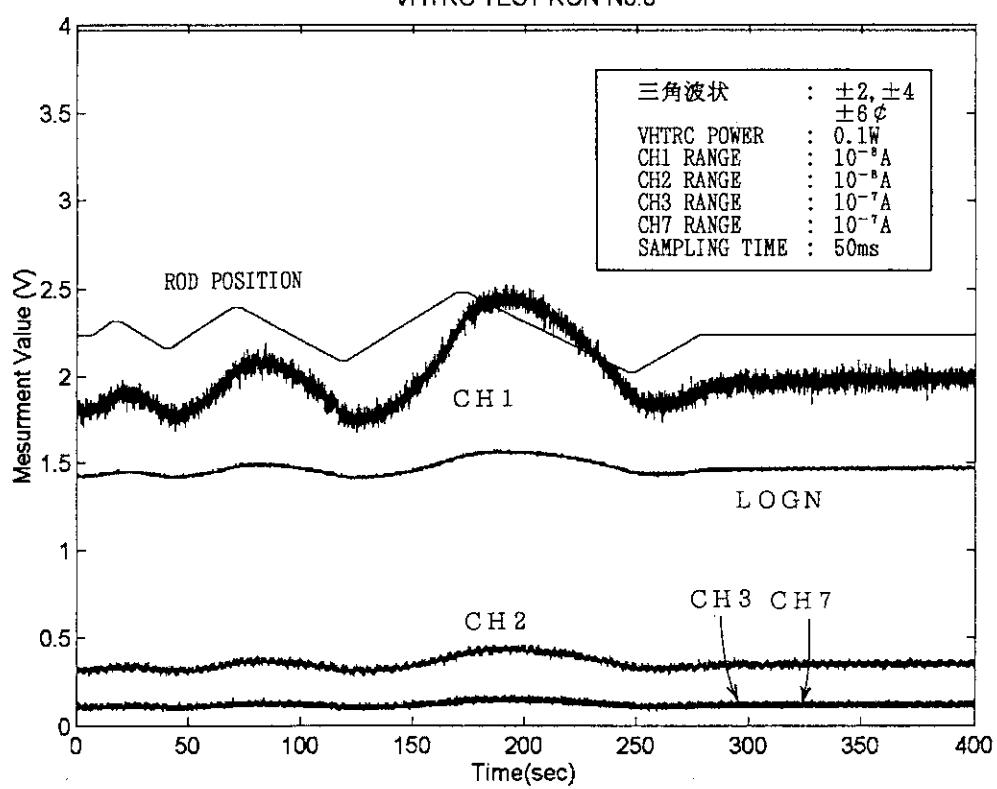
第4.8図 VHTRC実験データ（実験No.6）

VHTRC TEST RUN No.7

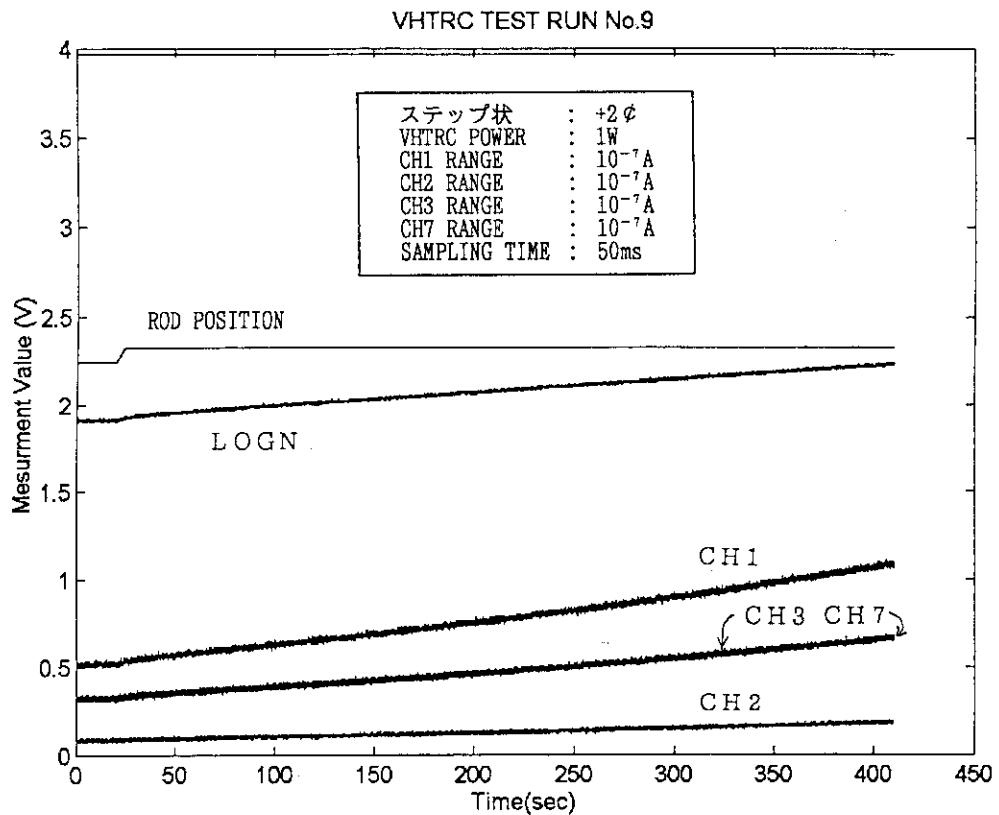


第4.9図 VHTRC実験データ（実験No.7）

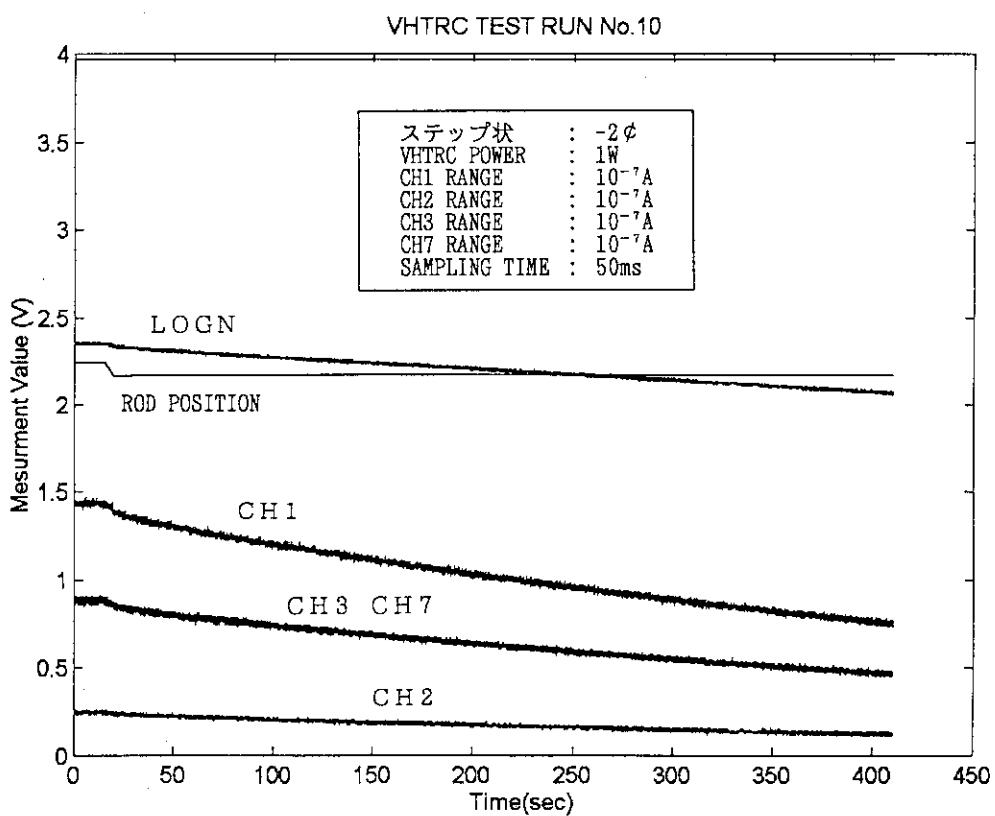
VHTRC TEST RUN No.8



第4.10図 VHTRC実験データ（実験No.8）

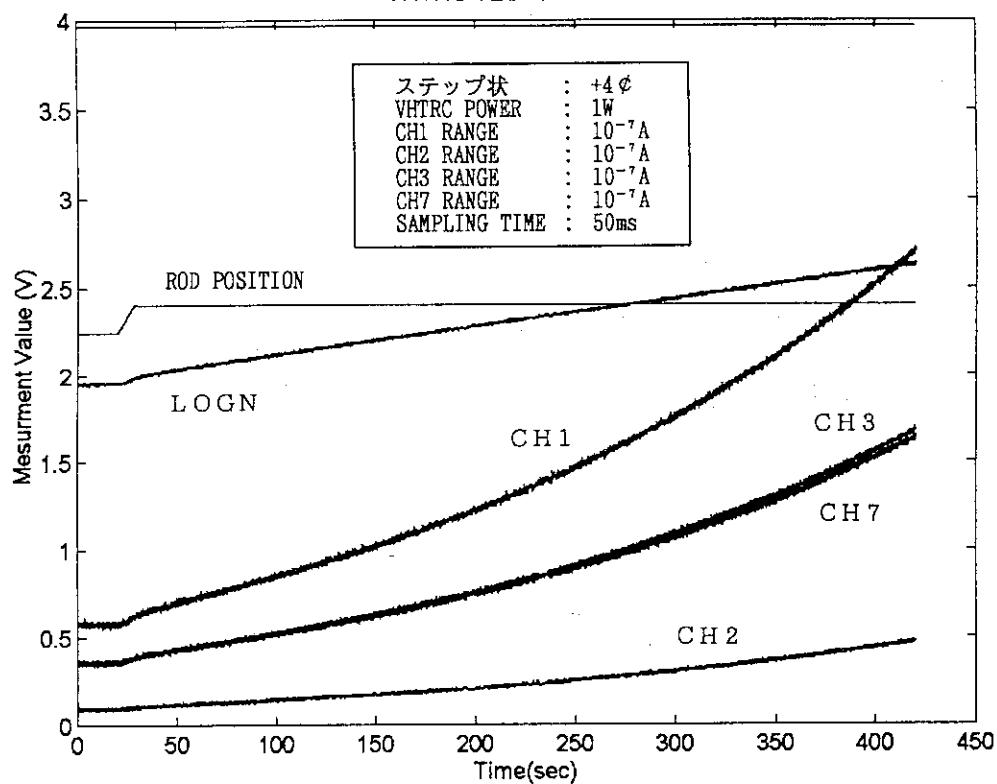


第4.11図 VHTRC実験データ（実験No.9）

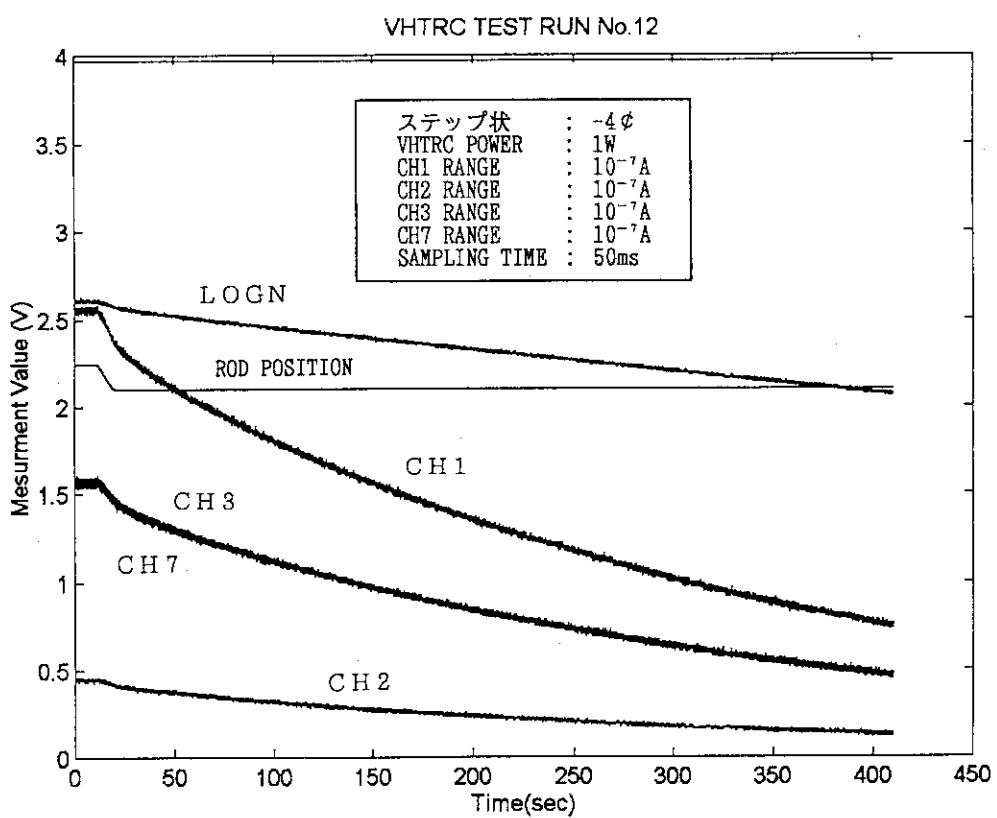


第4.12図 VHTRC実験データ（実験No.10）

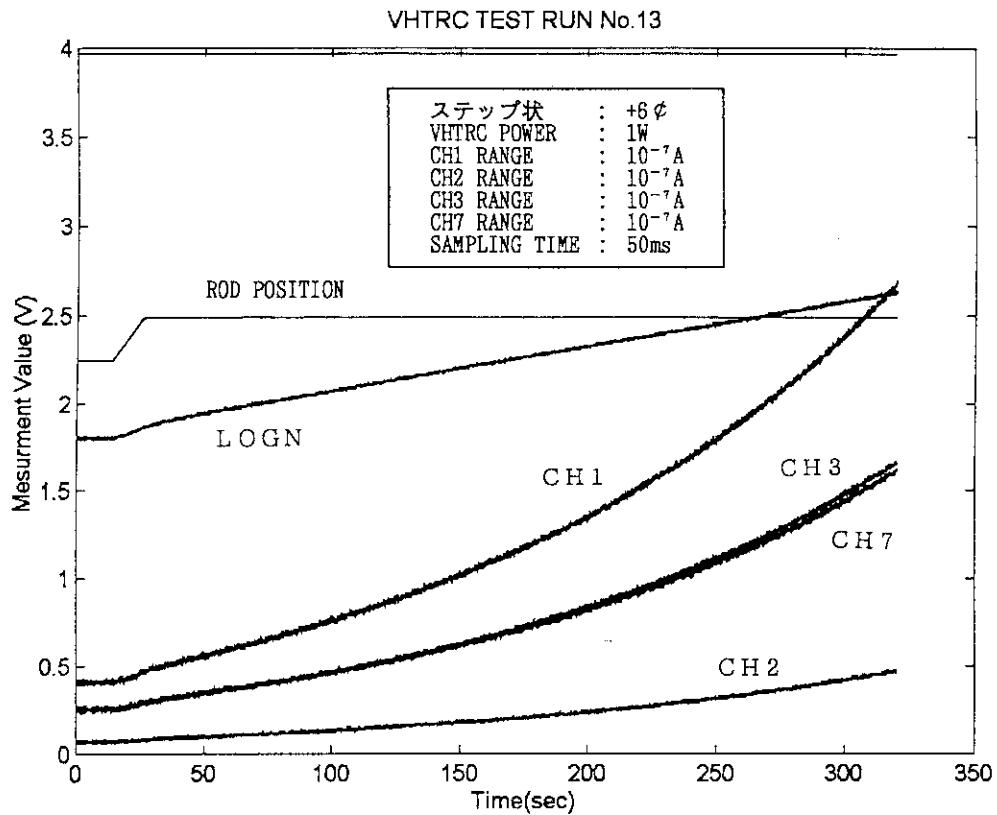
VHTRC TEST RUN No.11



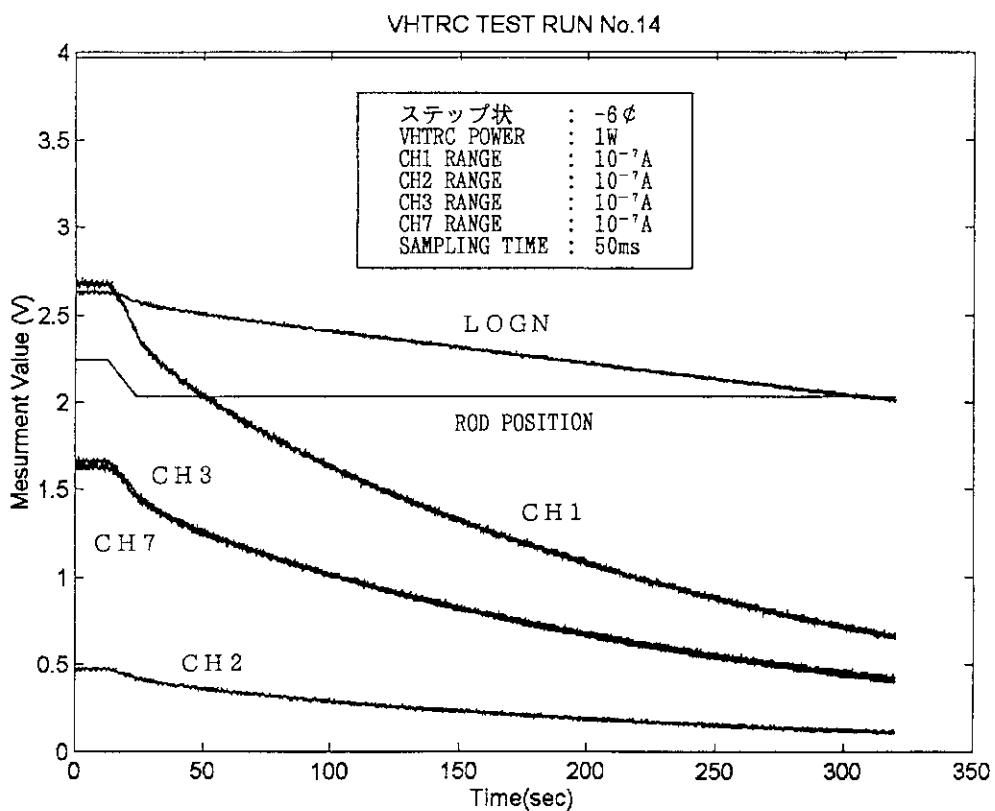
第4.13図 VHTRC実験データ（実験No.11）



第4.14図 VHTRC実験データ（実験No.12）

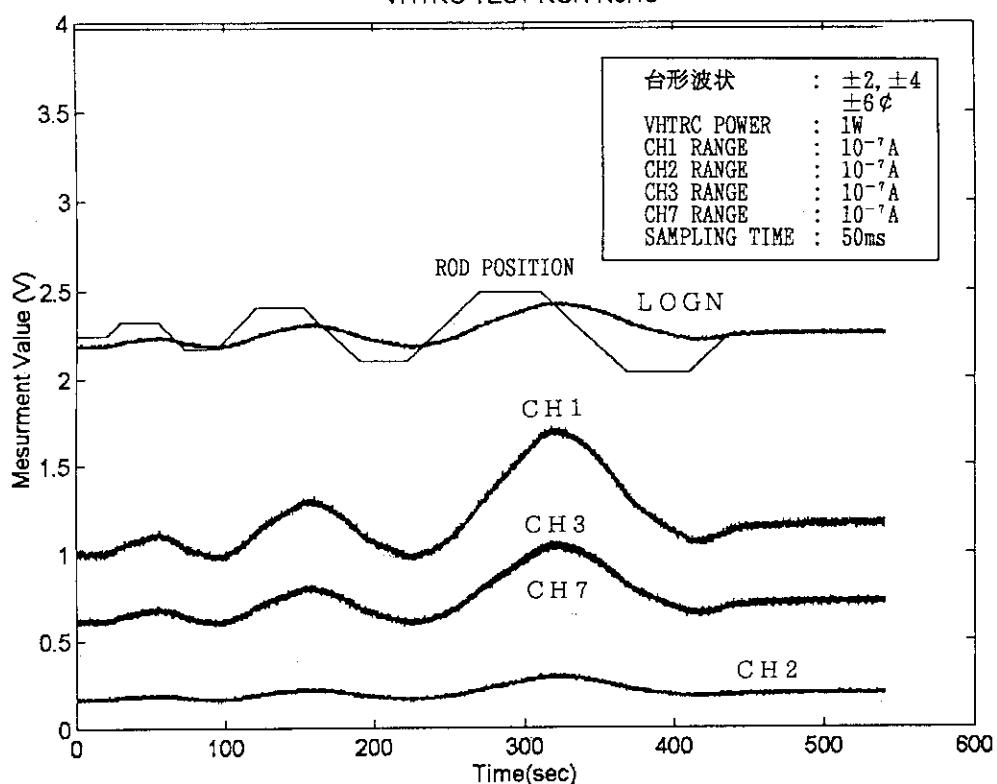


第4.15図 VHTRC実験データ（実験No.13）

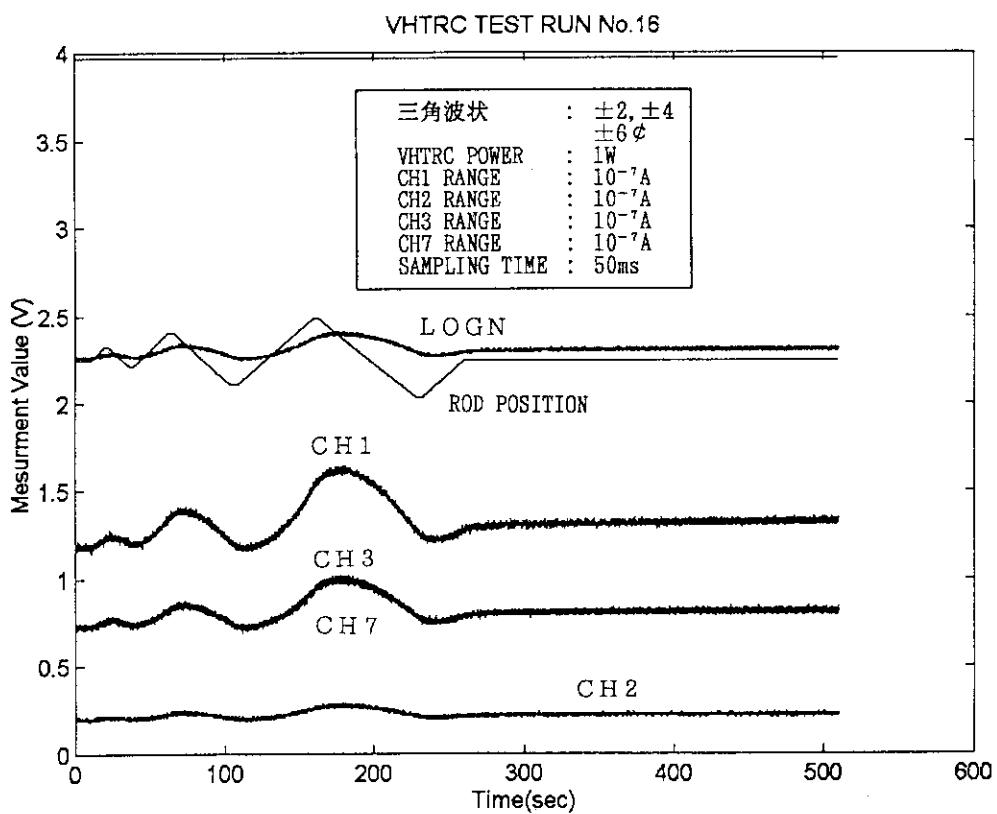


第4.16図 VHTRC実験データ（実験No.14）

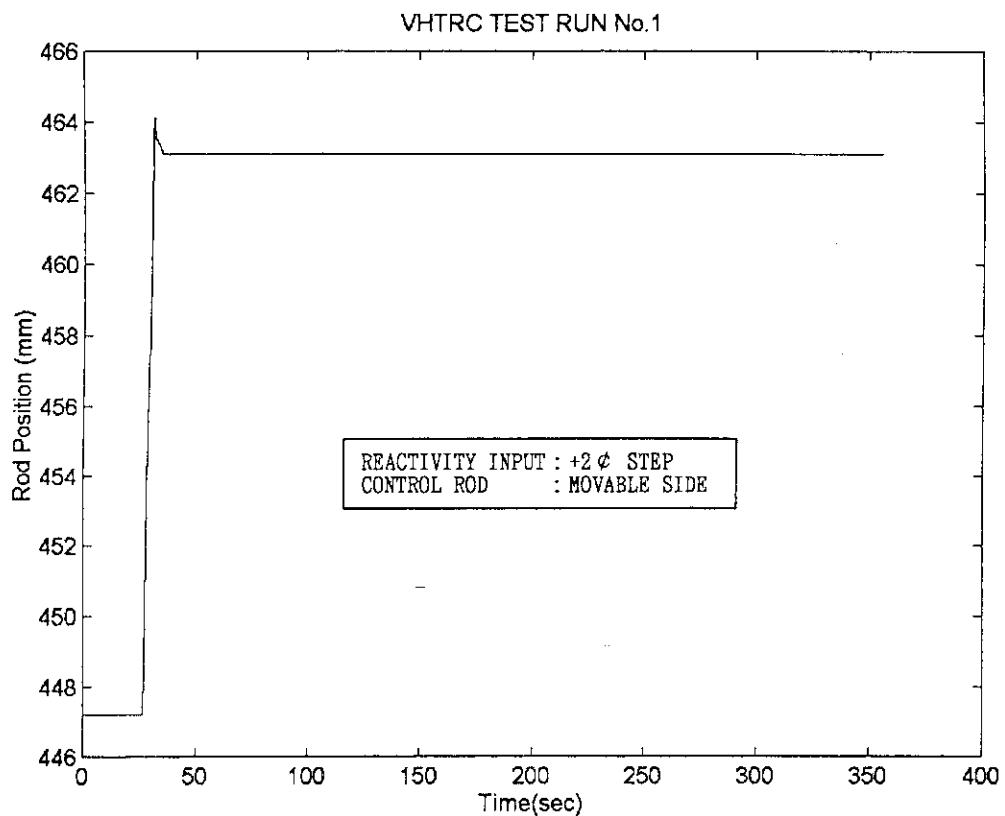
VHTRC TEST RUN No.15



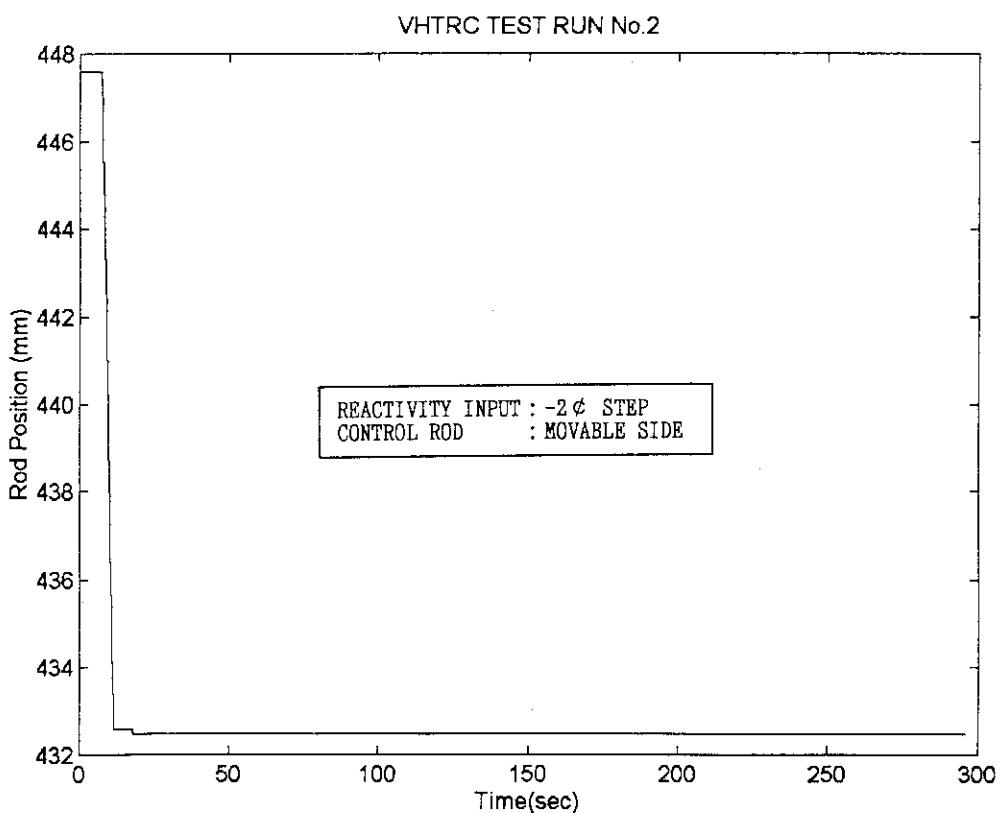
第4.17図 VHTRC実験データ（実験No.15）



第4.18図 VHTRC実験データ（実験No.16）

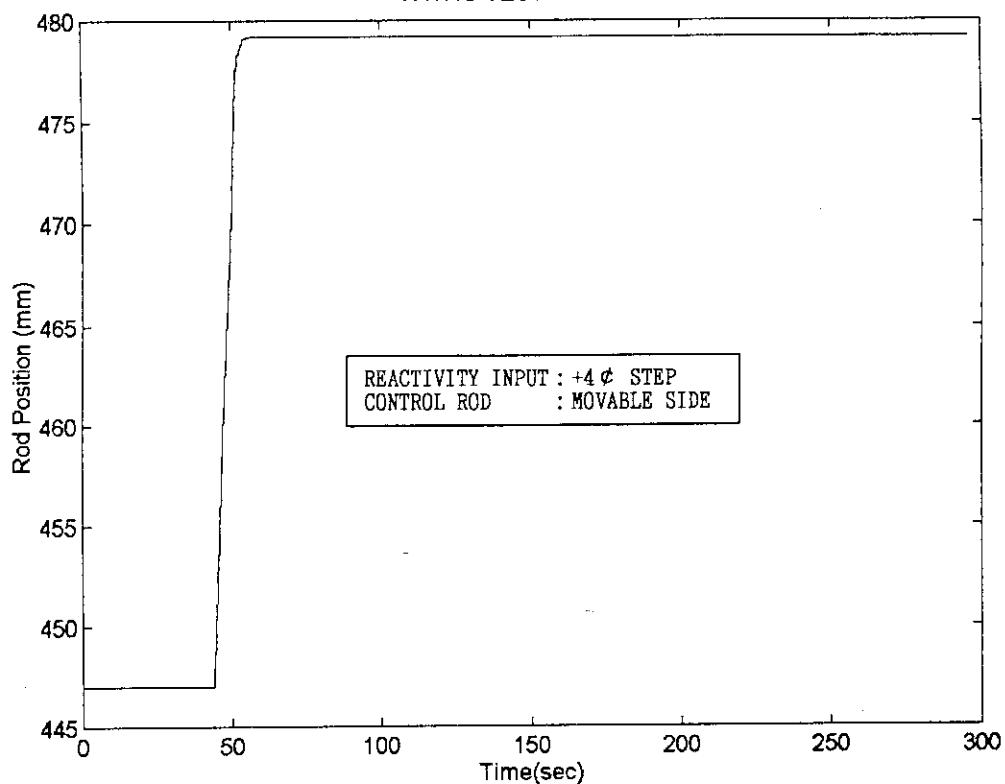


第4.19図 制御棒位置信号（実験No.1）



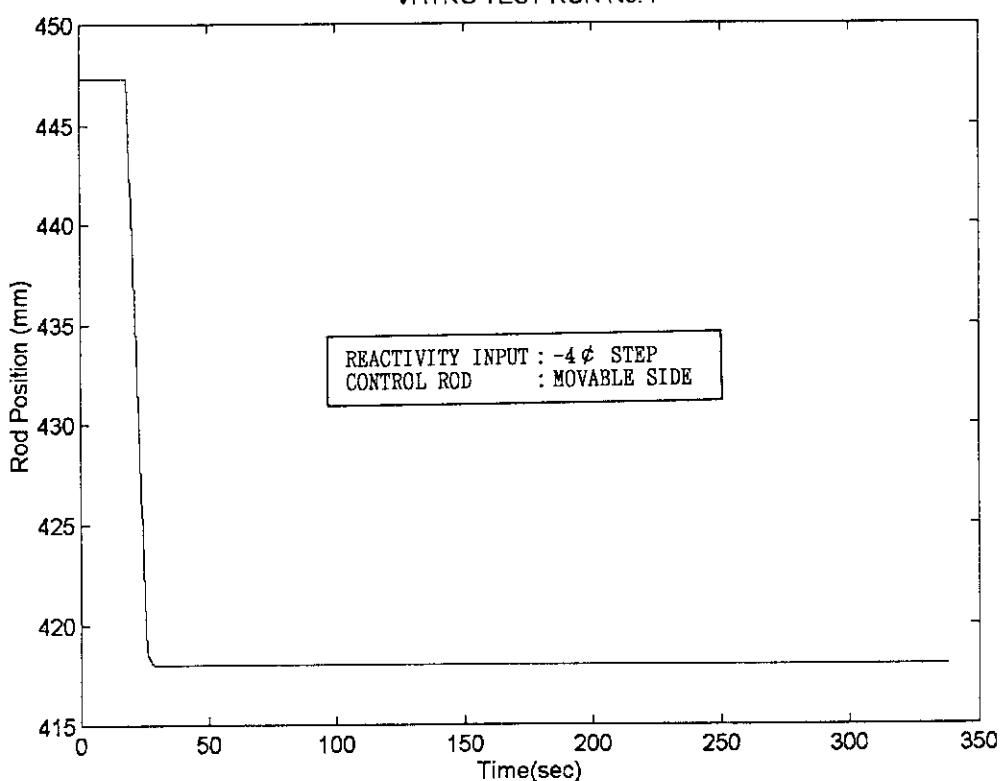
第4.20図 制御棒位置信号（実験No.2）

VHTRC TEST RUN No.3



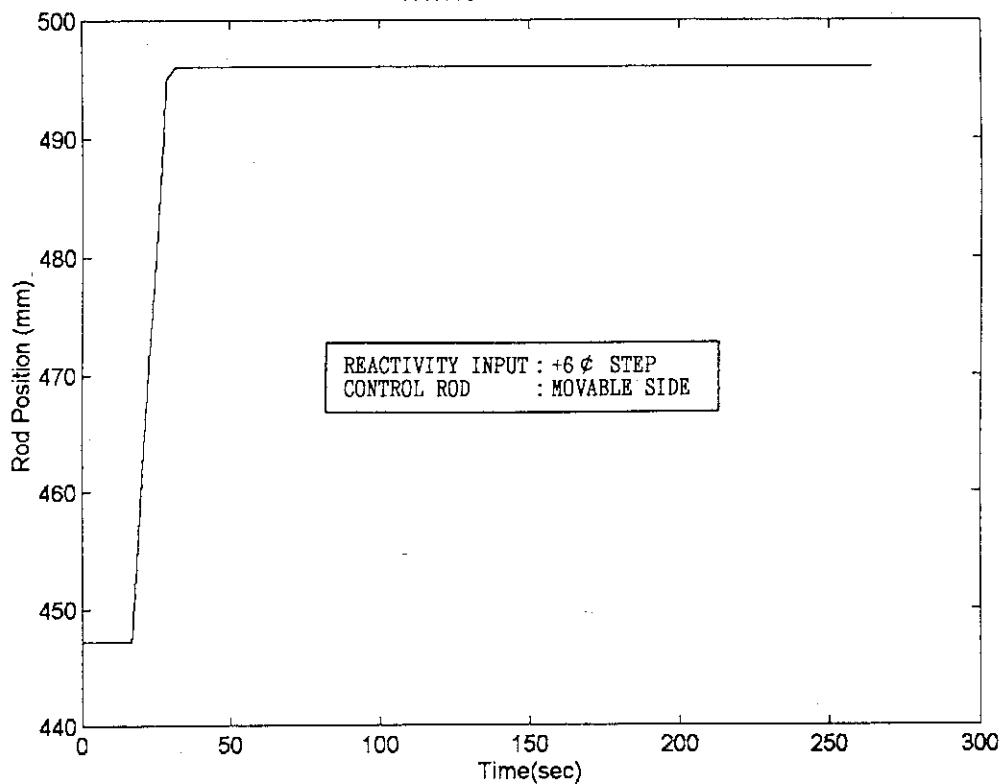
第4.21図 制御棒位置信号（実験No.3）

VHTRC TEST RUN No.4



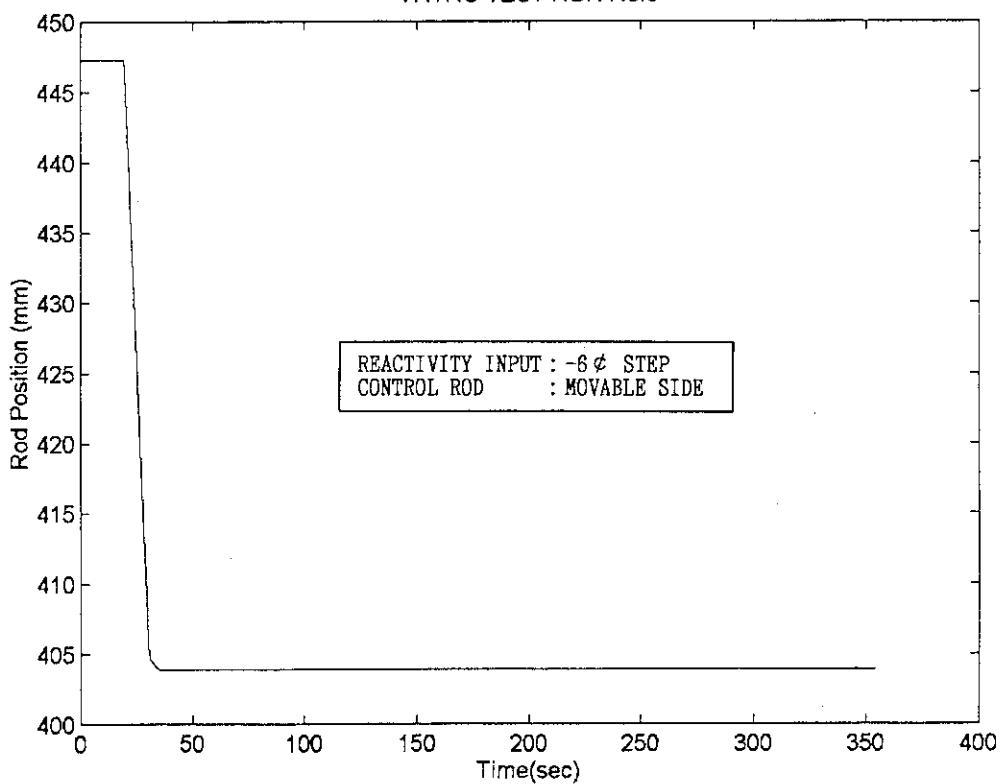
第4.22図 制御棒位置信号（実験No.4）

VHTRC TEST RUN No.5



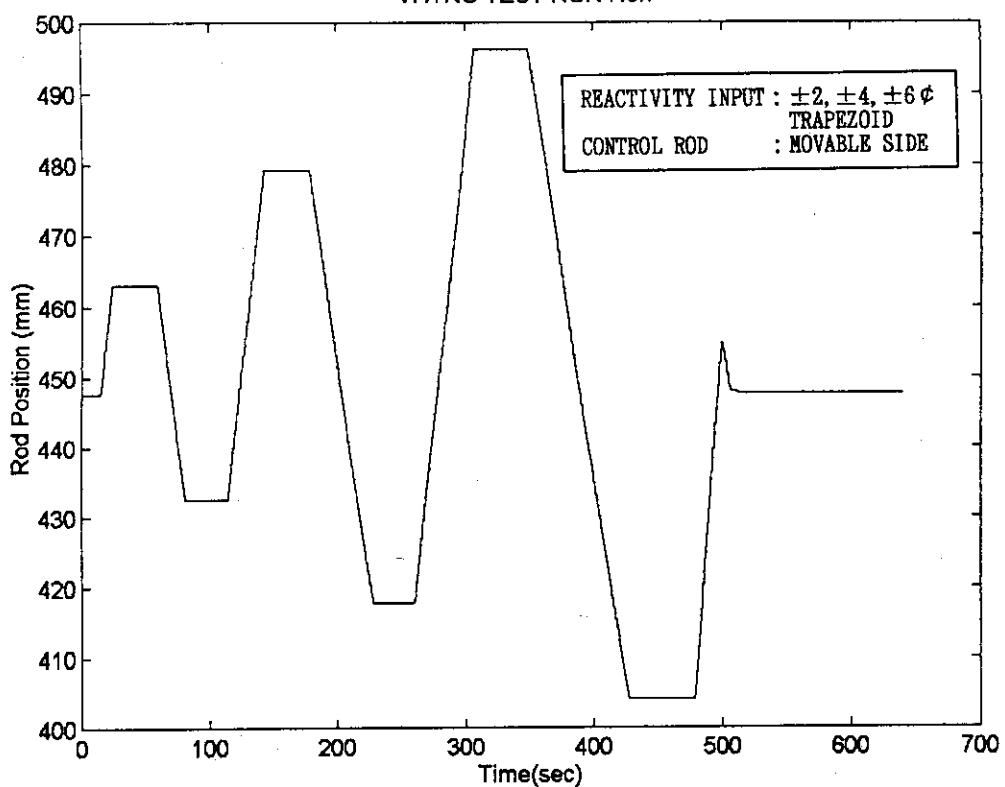
第4.23図 制御棒位置信号（実験No.5）

VHTRC TEST RUN No.6



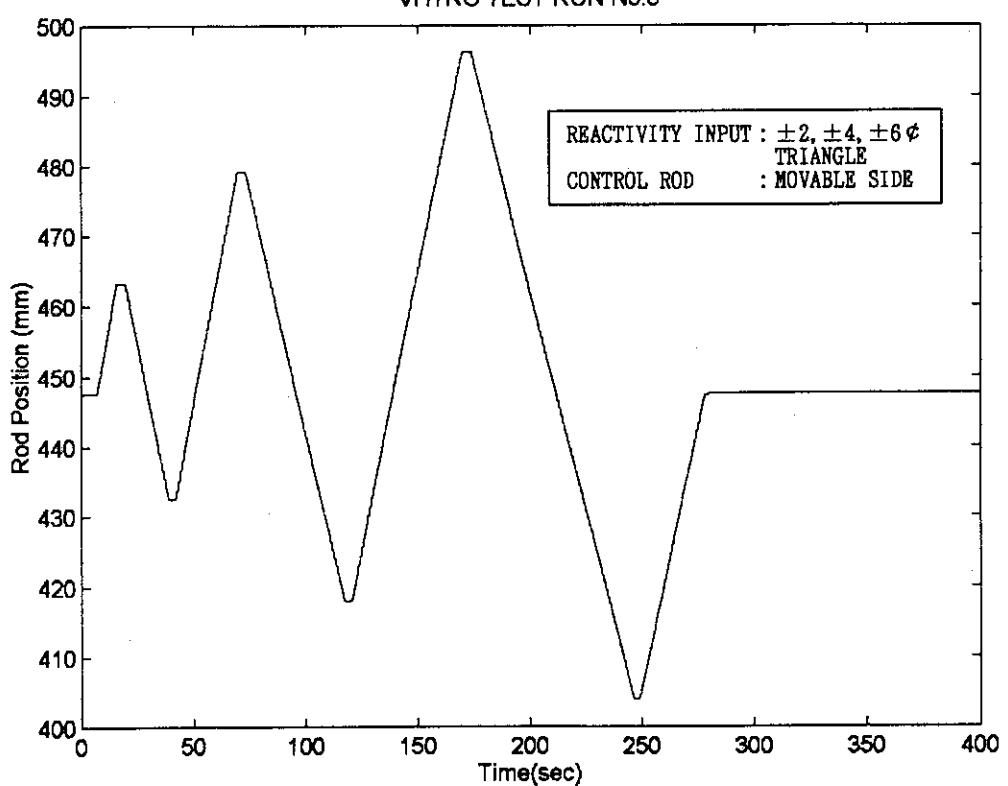
第4.24図 制御棒位置信号（実験No.6）

VHTRC TEST RUN No.7

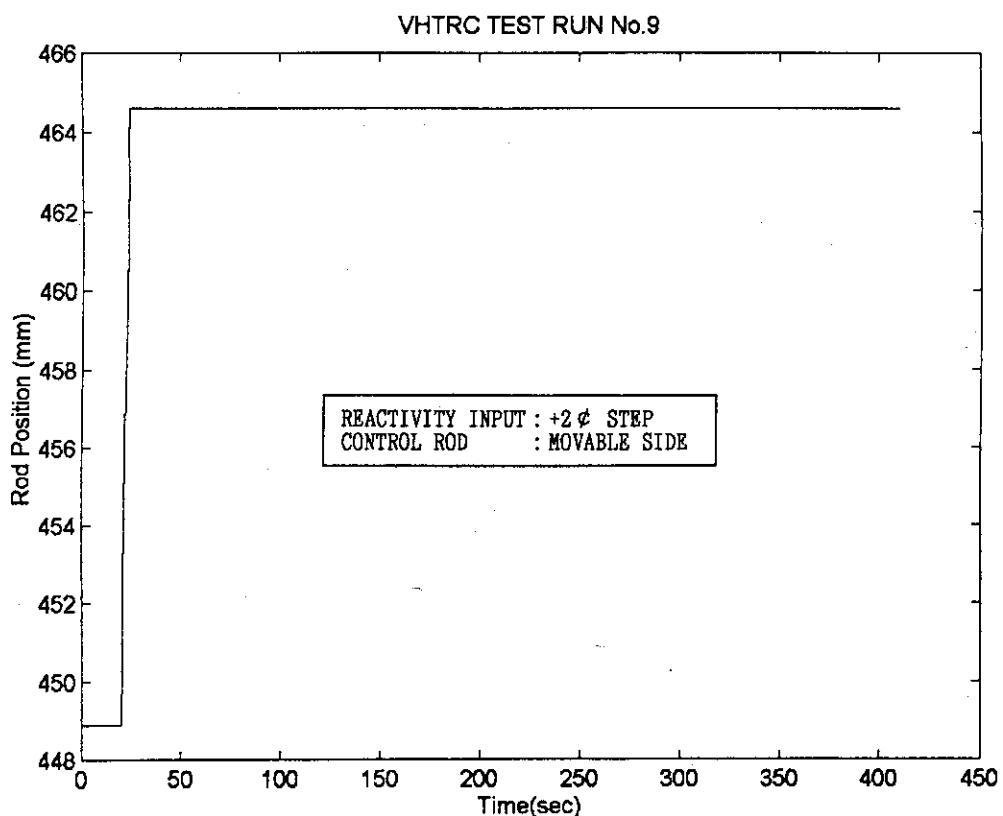


第4.25図 制御棒位置信号（実験No.7）

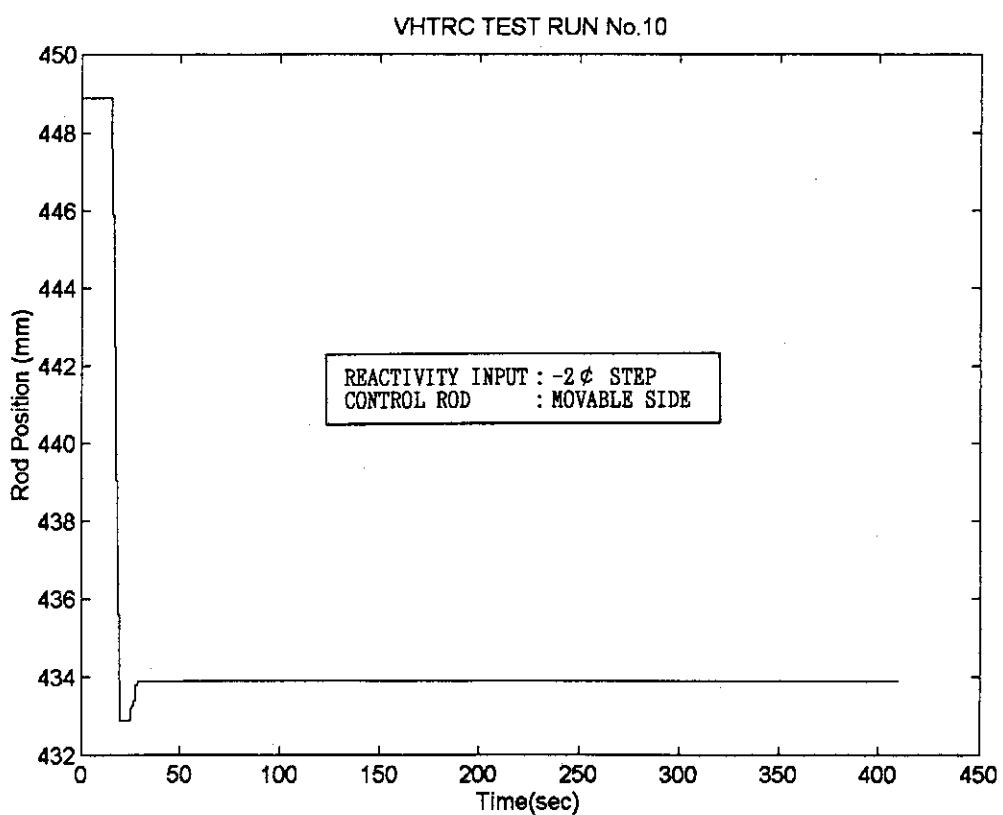
VHTRC TEST RUN No.8



第4.26図 制御棒位置信号（実験No.8）

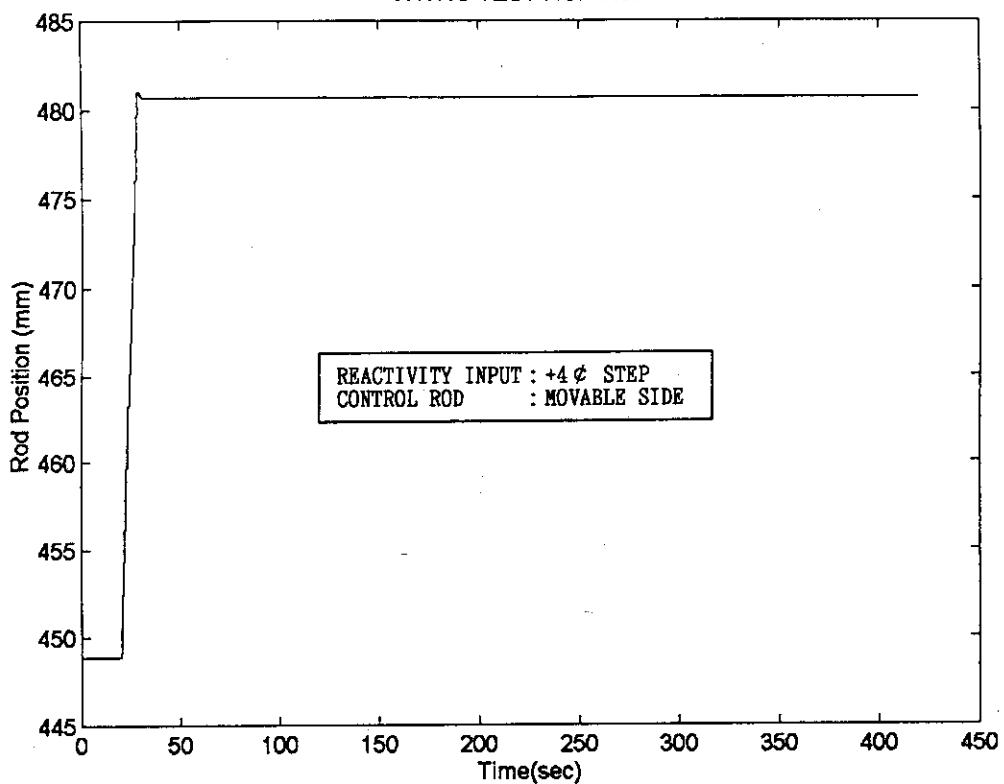


第4.27図 制御棒位置信号（実験No.9）



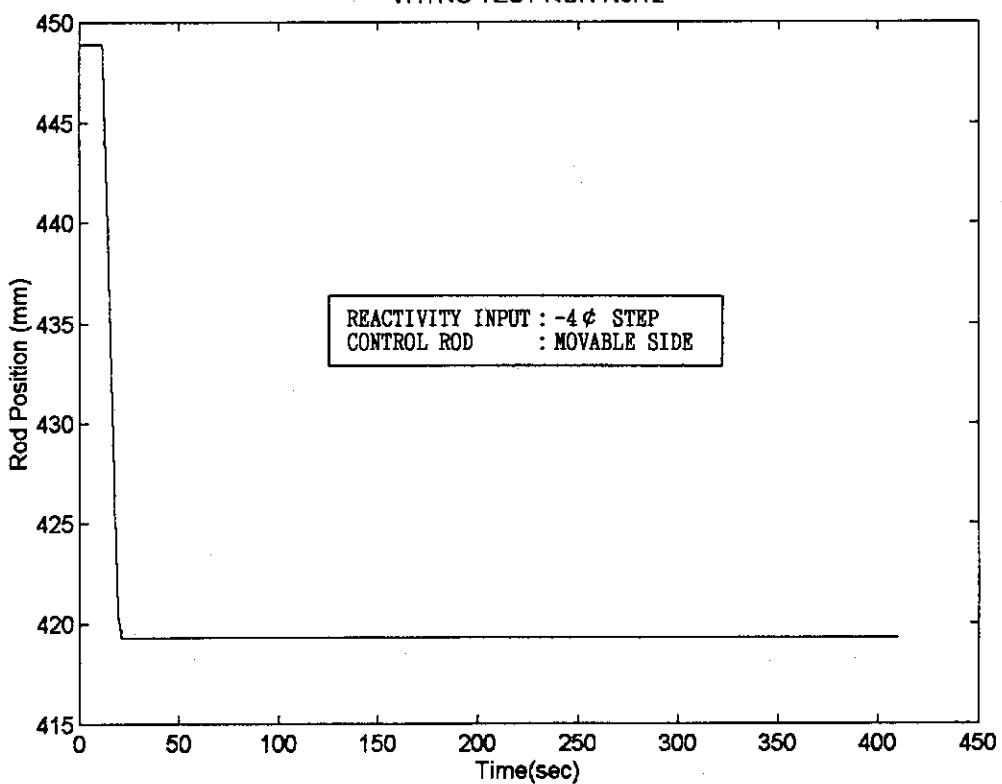
第4.28図 制御棒位置信号（実験No.10）

VHTRC TEST RUN No.11



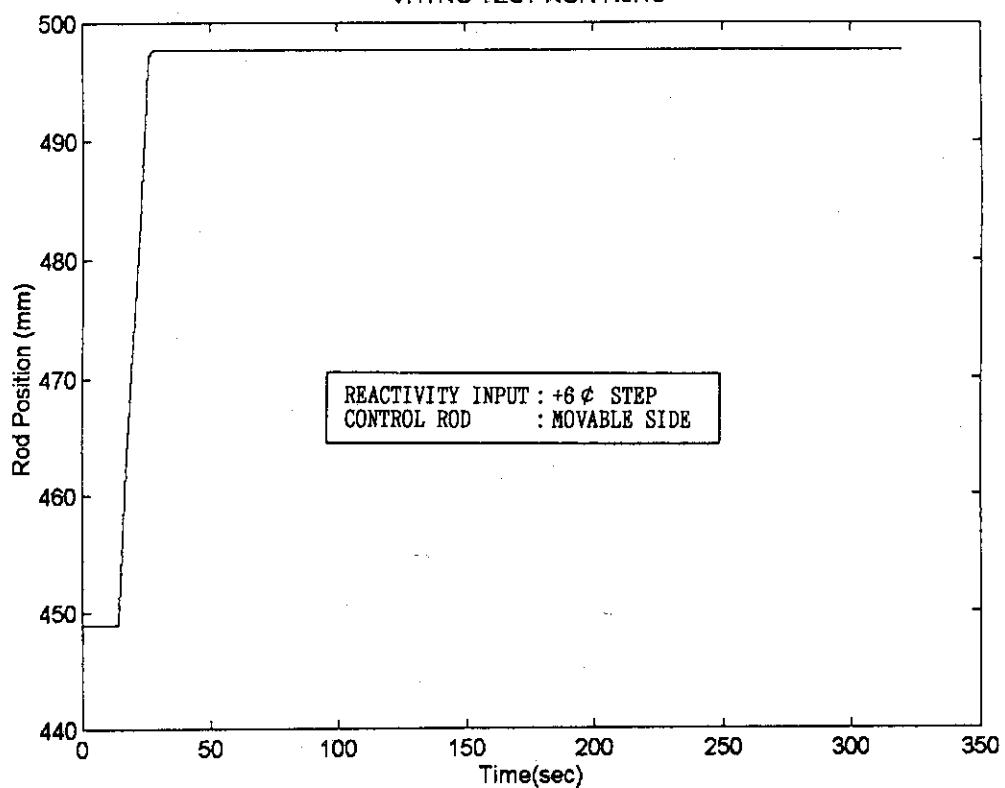
第4.29図 制御棒位置信号（実験No.11）

VHTRC TEST RUN No.12



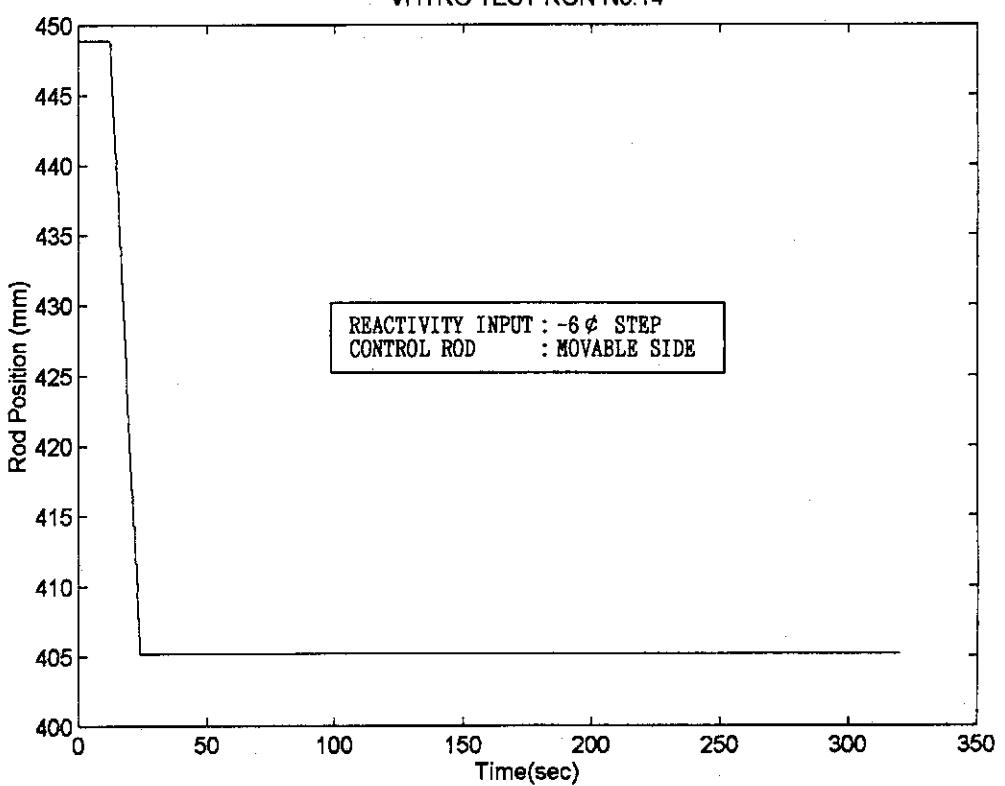
第4.30図 制御棒位置信号（実験No.12）

VHTRC TEST RUN No.13



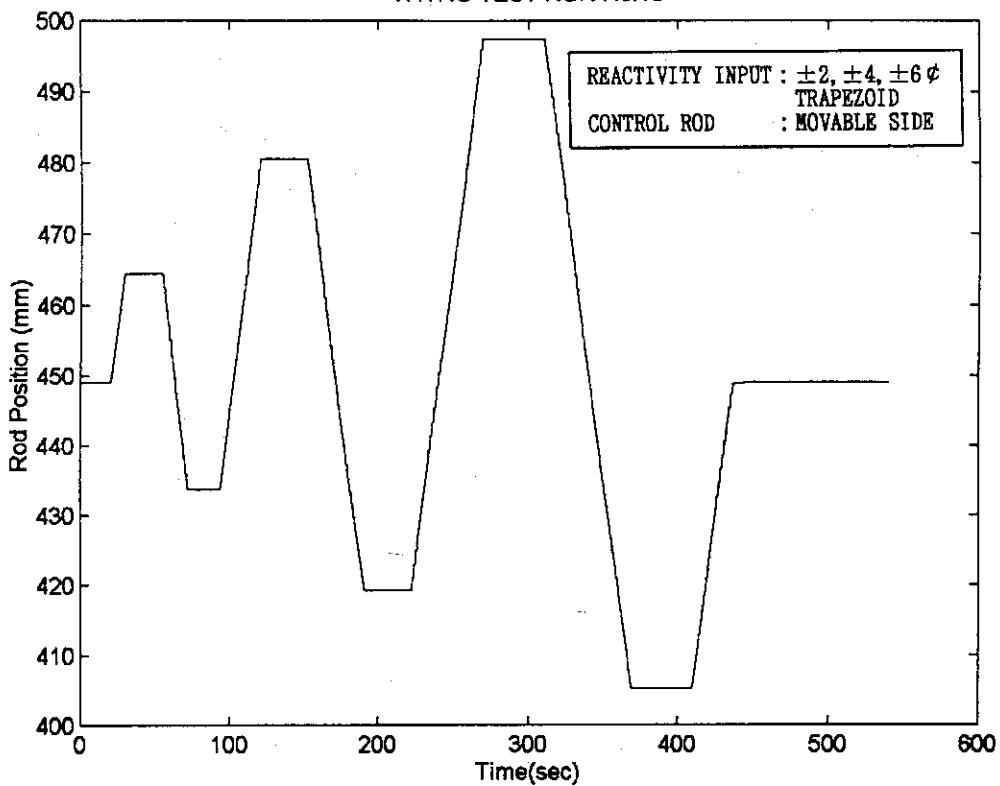
第4.31図 制御棒位置信号（実験No.13）

VHTRC TEST RUN No.14



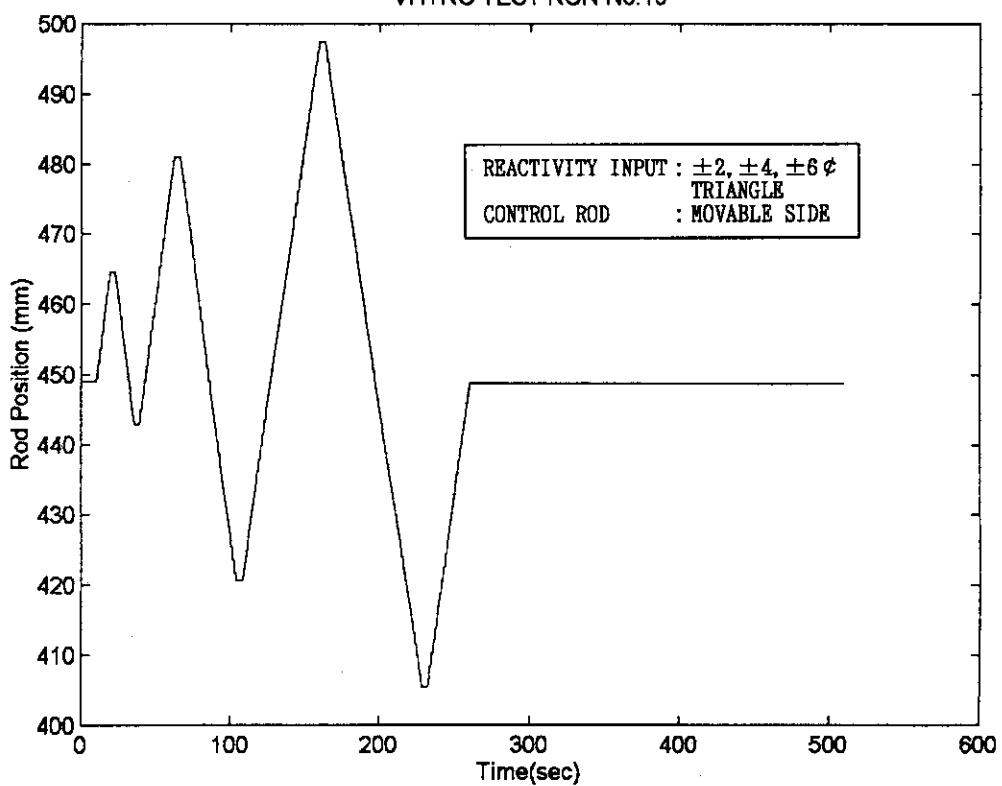
第4.32図 制御棒位置信号（実験No.14）

VHTRC TEST RUN No.15



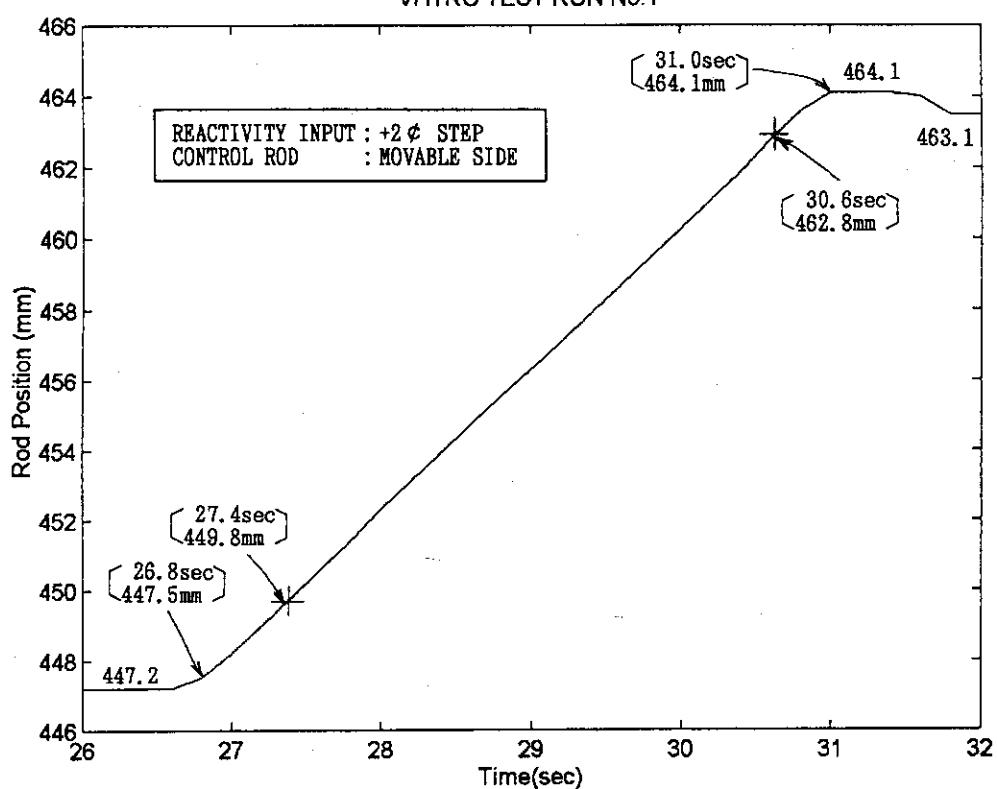
第4.33図 制御棒位置信号（実験No.15）

VHTRC TEST RUN No.16



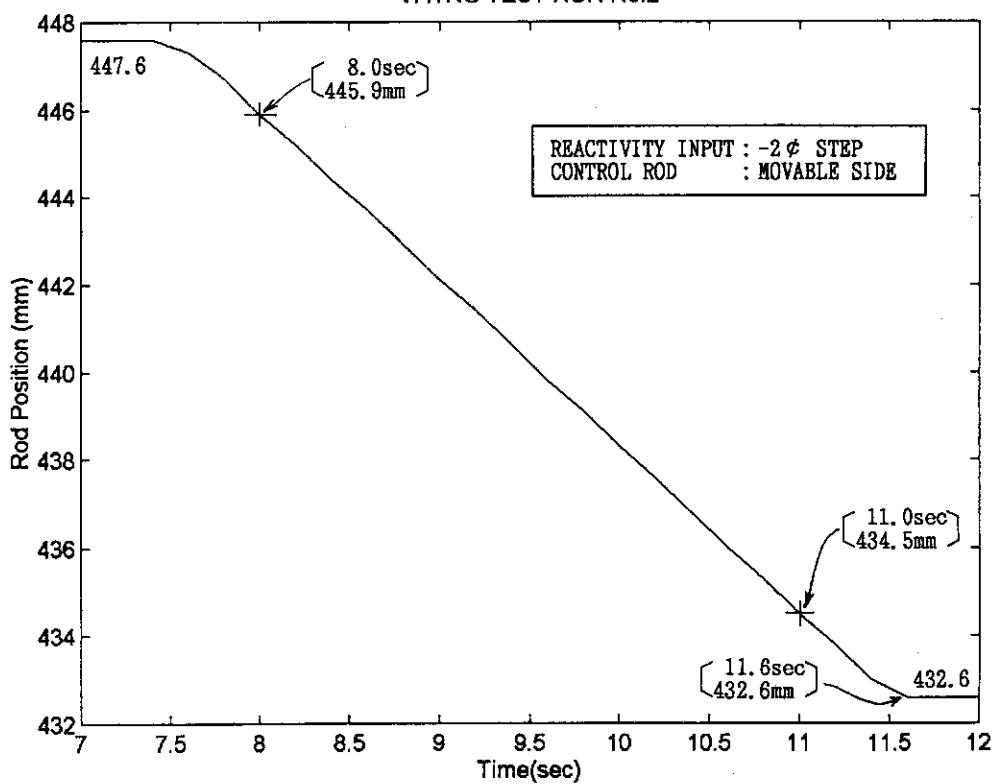
第4.34図 制御棒位置信号（実験No.16）

VHTRC TEST RUN No.1

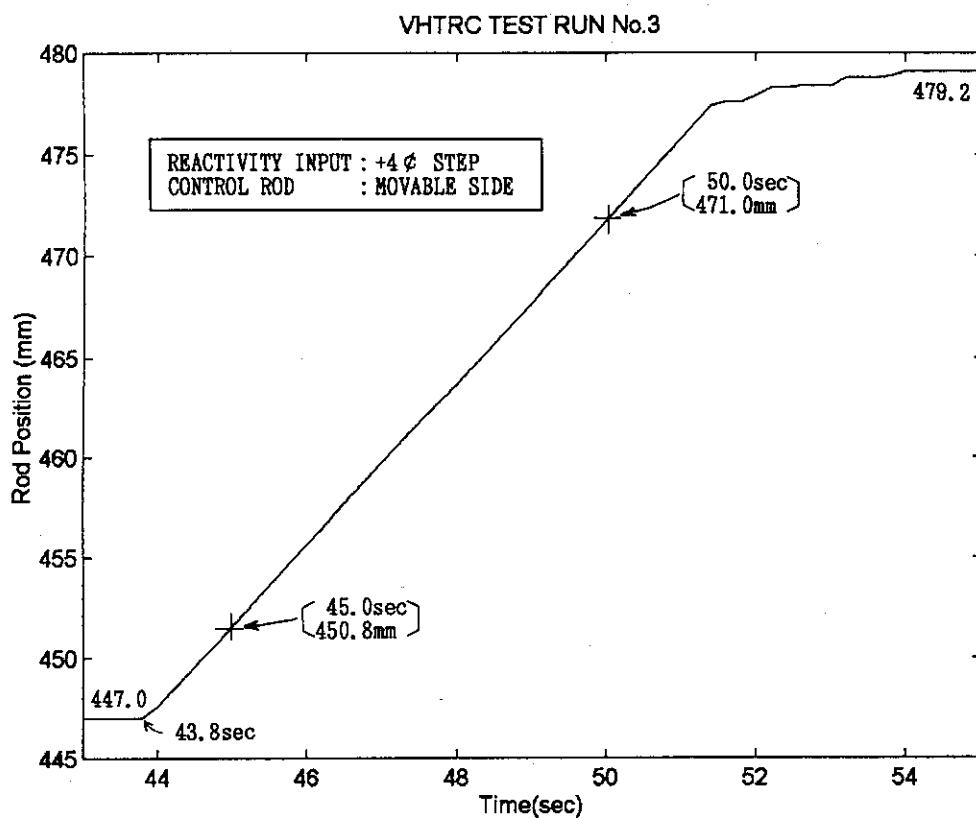


第4.35図 制御棒位置信号トランジェント部分の拡大図（実験No.1）

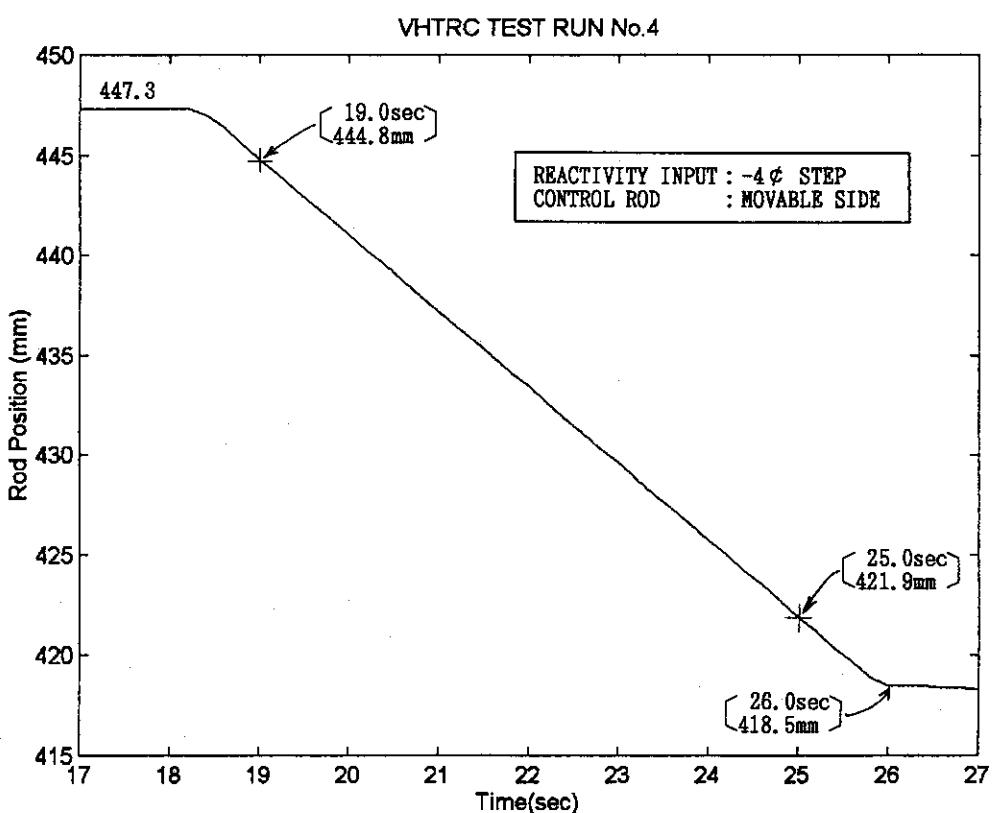
VHTRC TEST RUN No.2



第4.36図 制御棒位置信号トランジェント部分の拡大図（実験No.2）

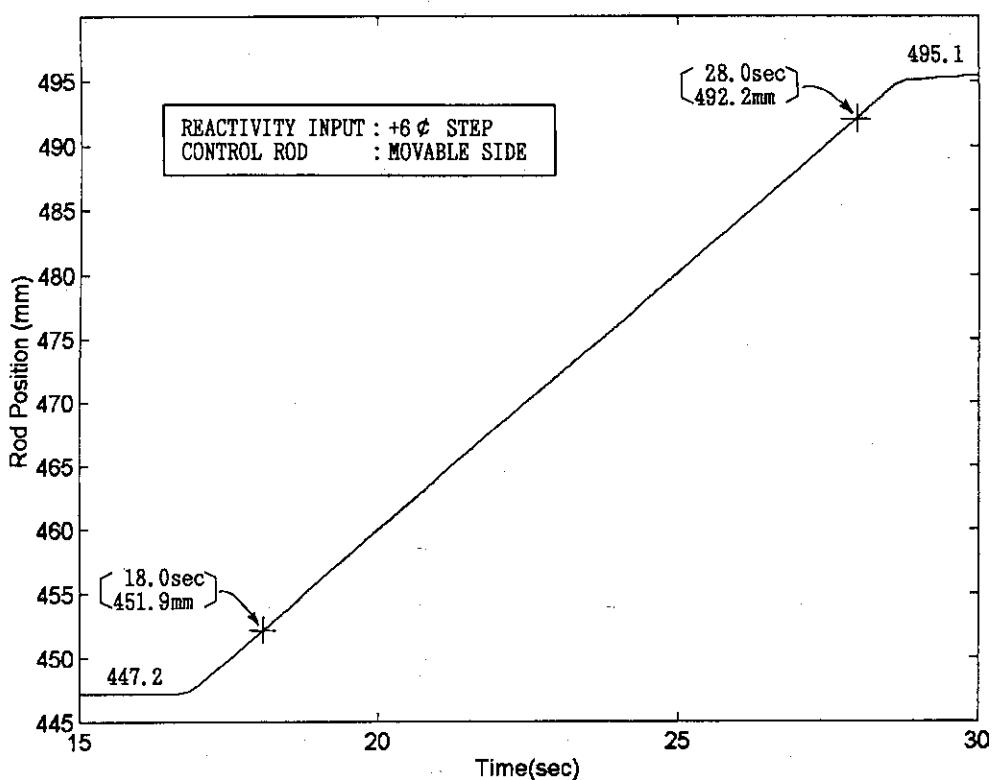


第4.37図 制御棒位置信号トランジエント部分の拡大図（実験No.3）



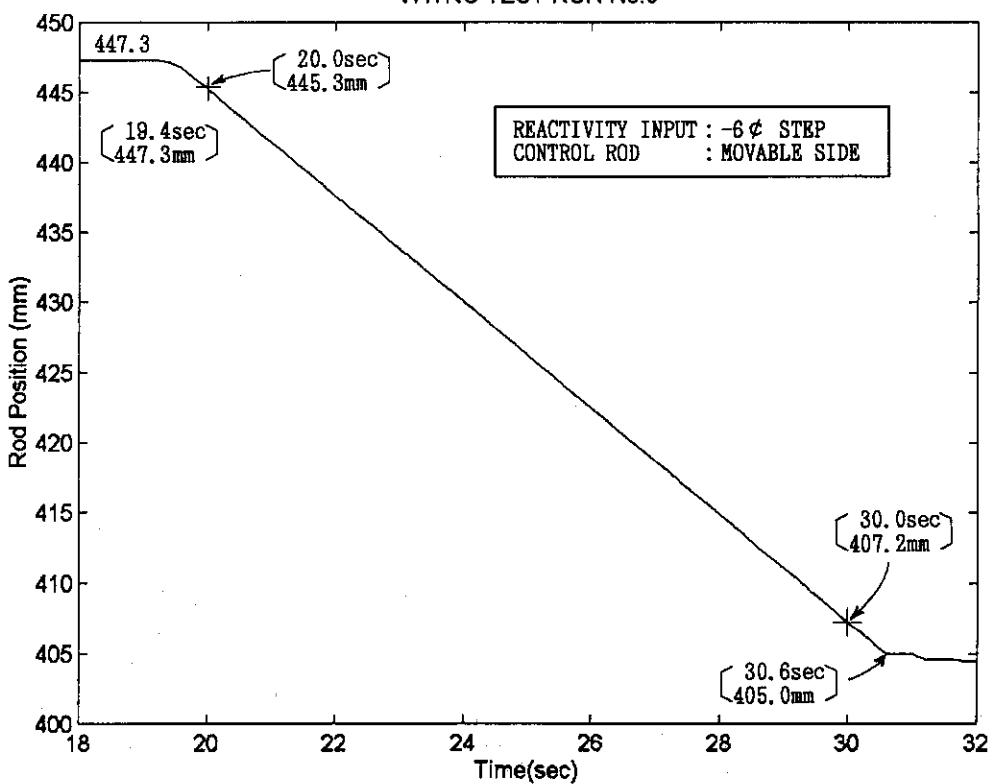
第4.38図 制御棒位置信号トランジエント部分の拡大図（実験No.4）

VHTRC TEST RUN No.5

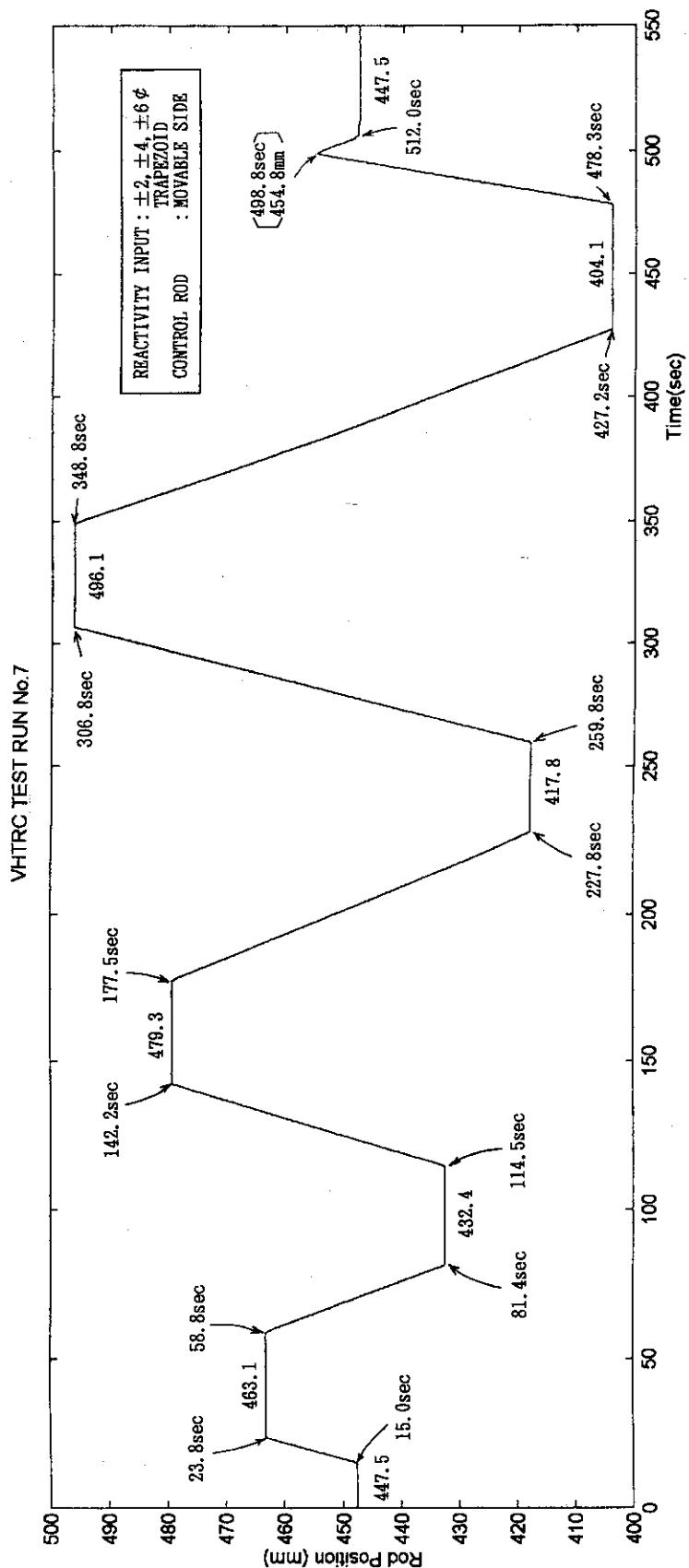


第4.39図 制御棒位置信号トランジェント部分の拡大図（実験No.5）

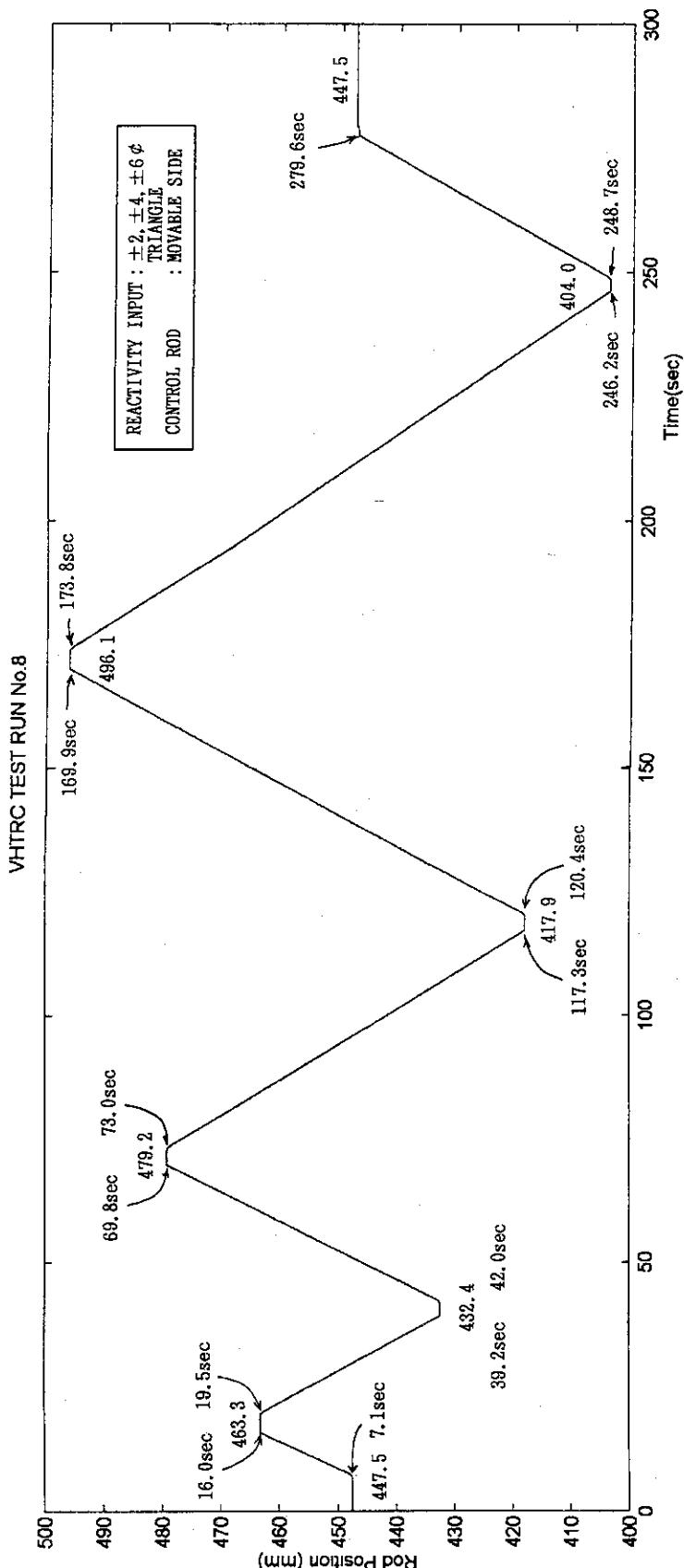
VHTRC TEST RUN No.6



第4.40図 制御棒位置信号トランジェント部分の拡大図（実験No.6）

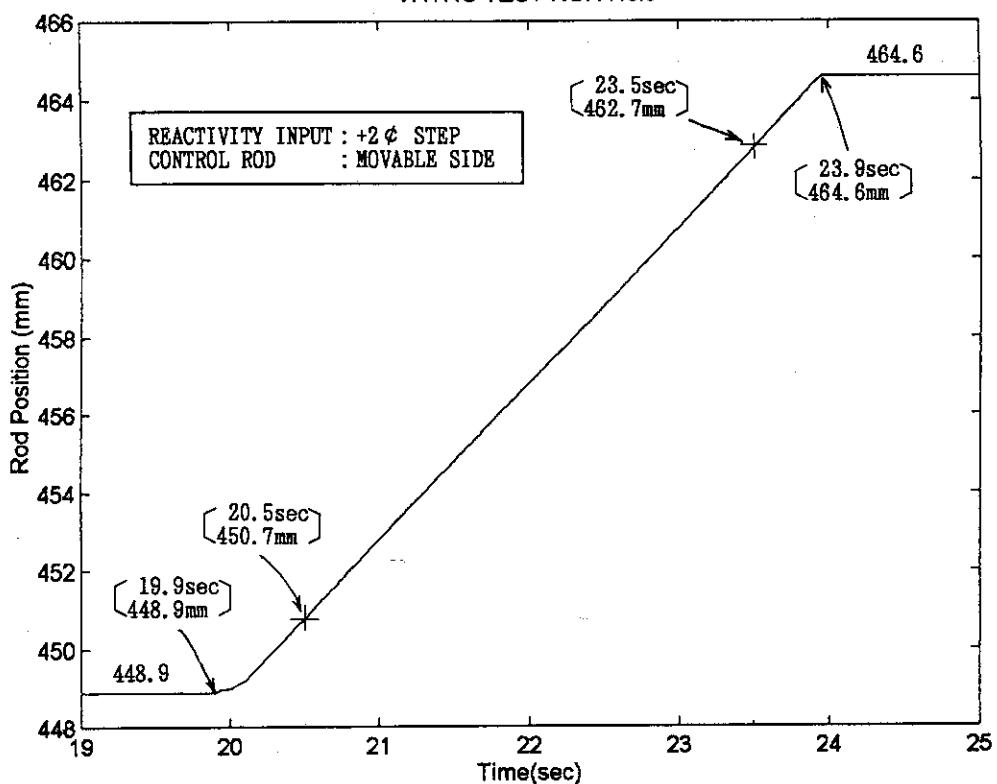


第4.41図 制御棒位置信号トランジエント部分の拡大図（実験No.7）



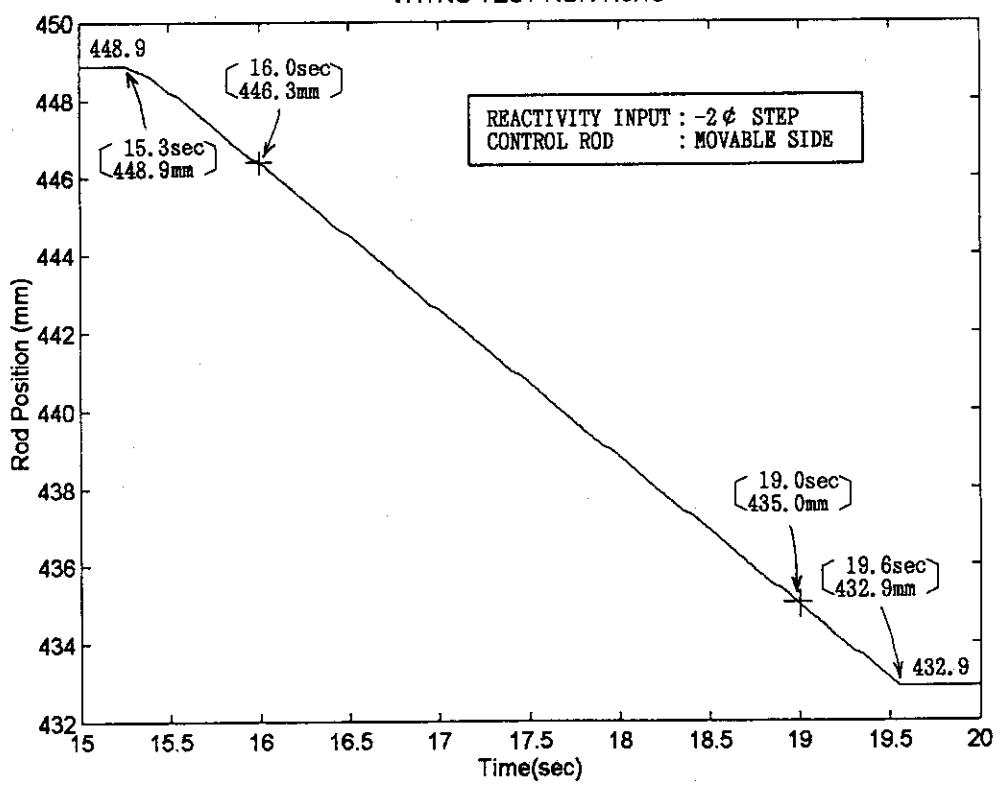
第4.42図 制御棒位置信号トランジメント部分の拡大図（実験No.8）

VHTRC TEST RUN No.9



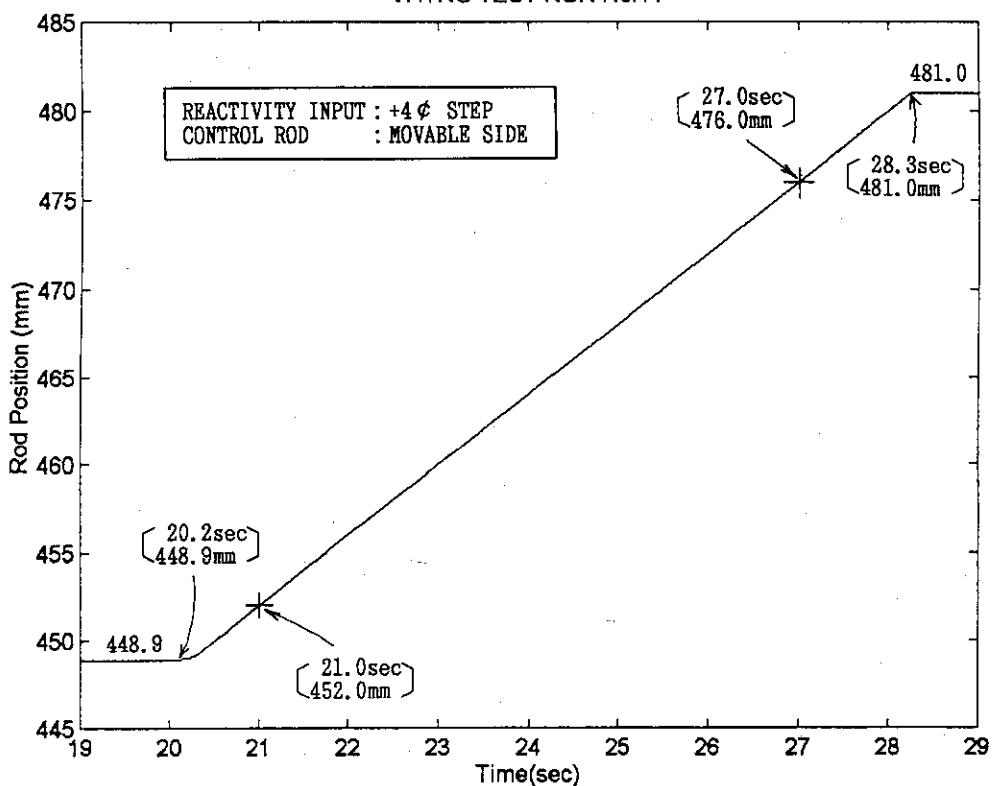
第4.43図 制御棒位置信号トランジエント部分の拡大図（実験No.9）

VHTRC TEST RUN No.10



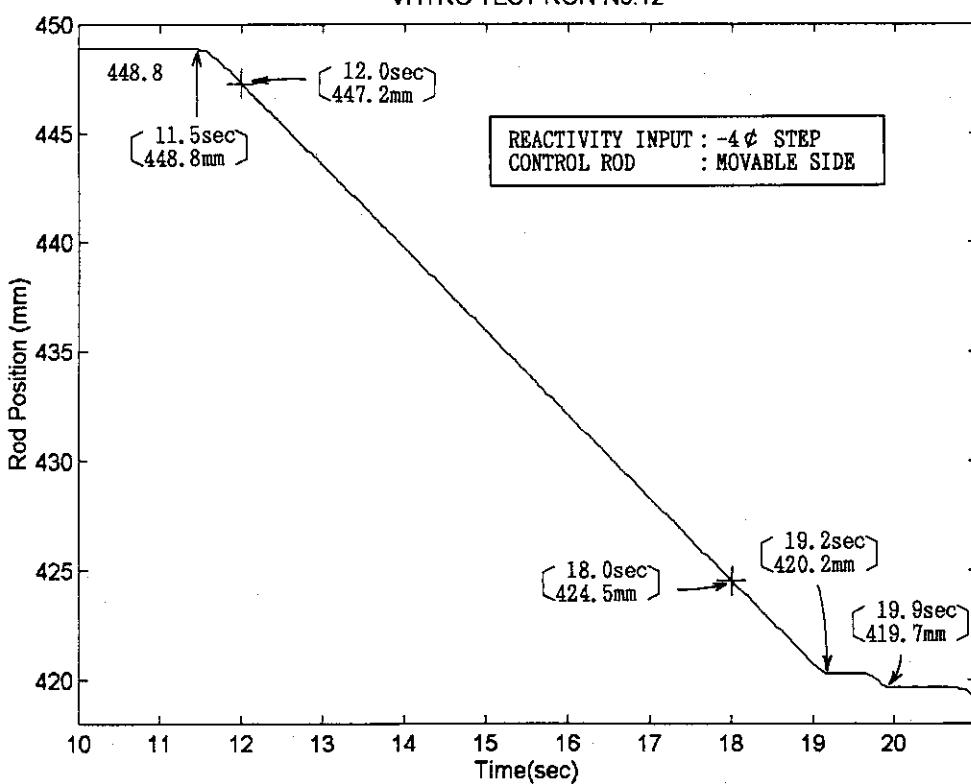
第4.44図 制御棒位置信号トランジエント部分の拡大図（実験No.10）

VHTRC TEST RUN No.11



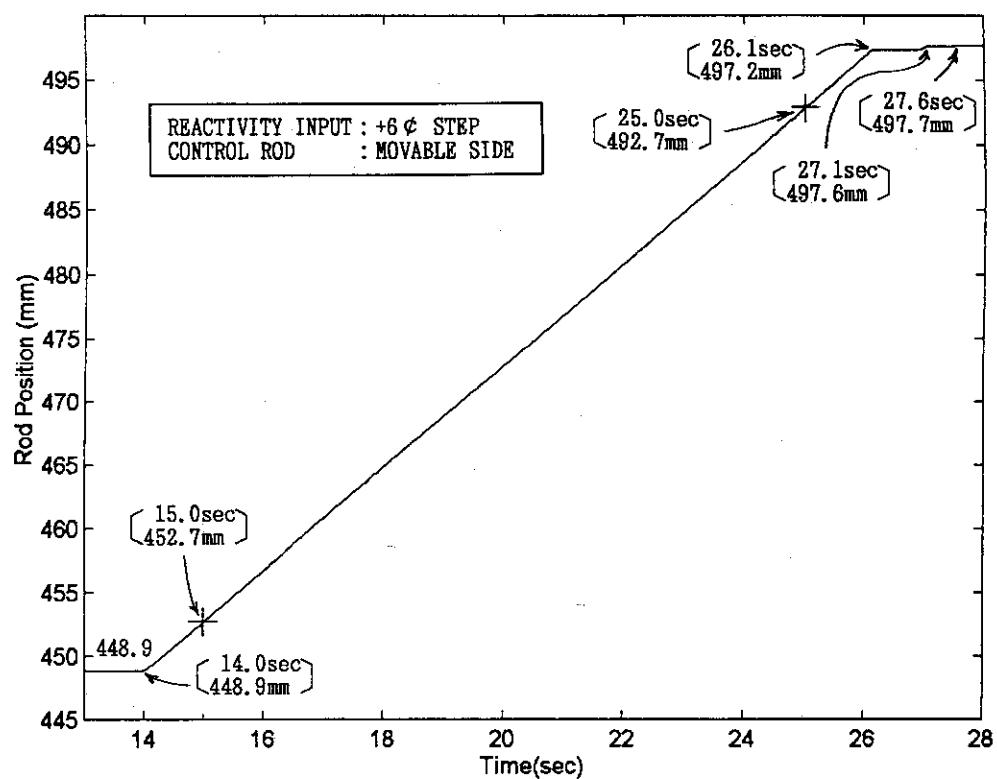
第4.45図 制御棒位置信号トランジエント部分の拡大図（実験No.11）

VHTRC TEST RUN No.12



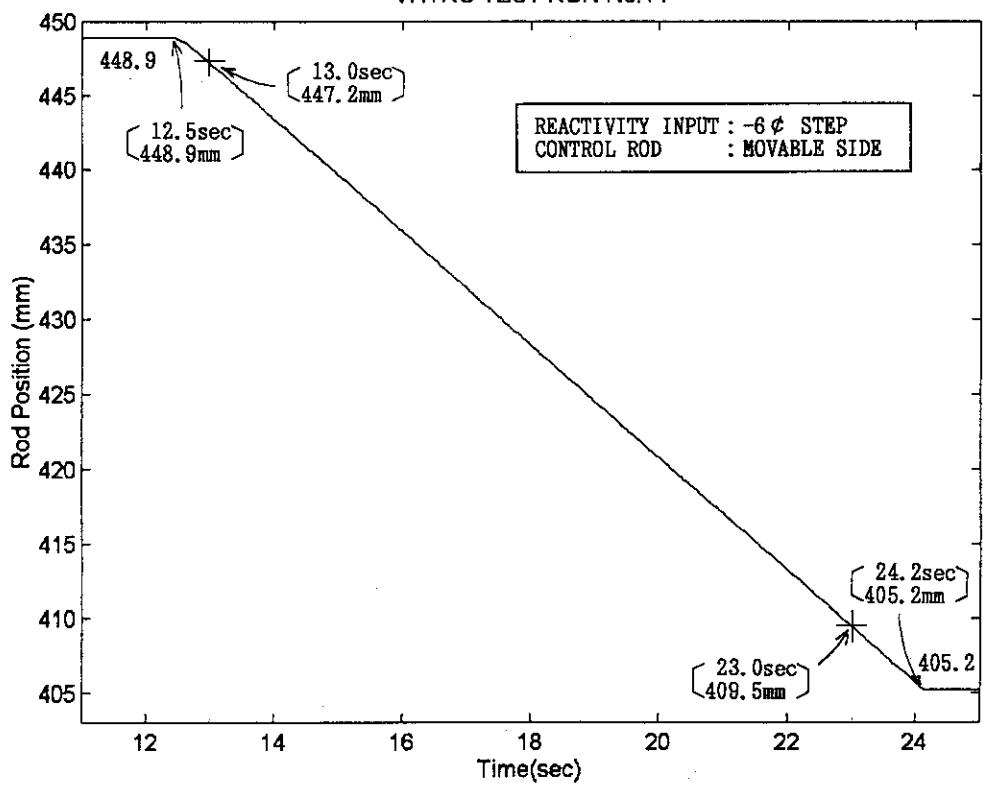
第4.46図 制御棒位置信号トランジエント部分の拡大図（実験No.12）

VHTRC TEST RUN No.13

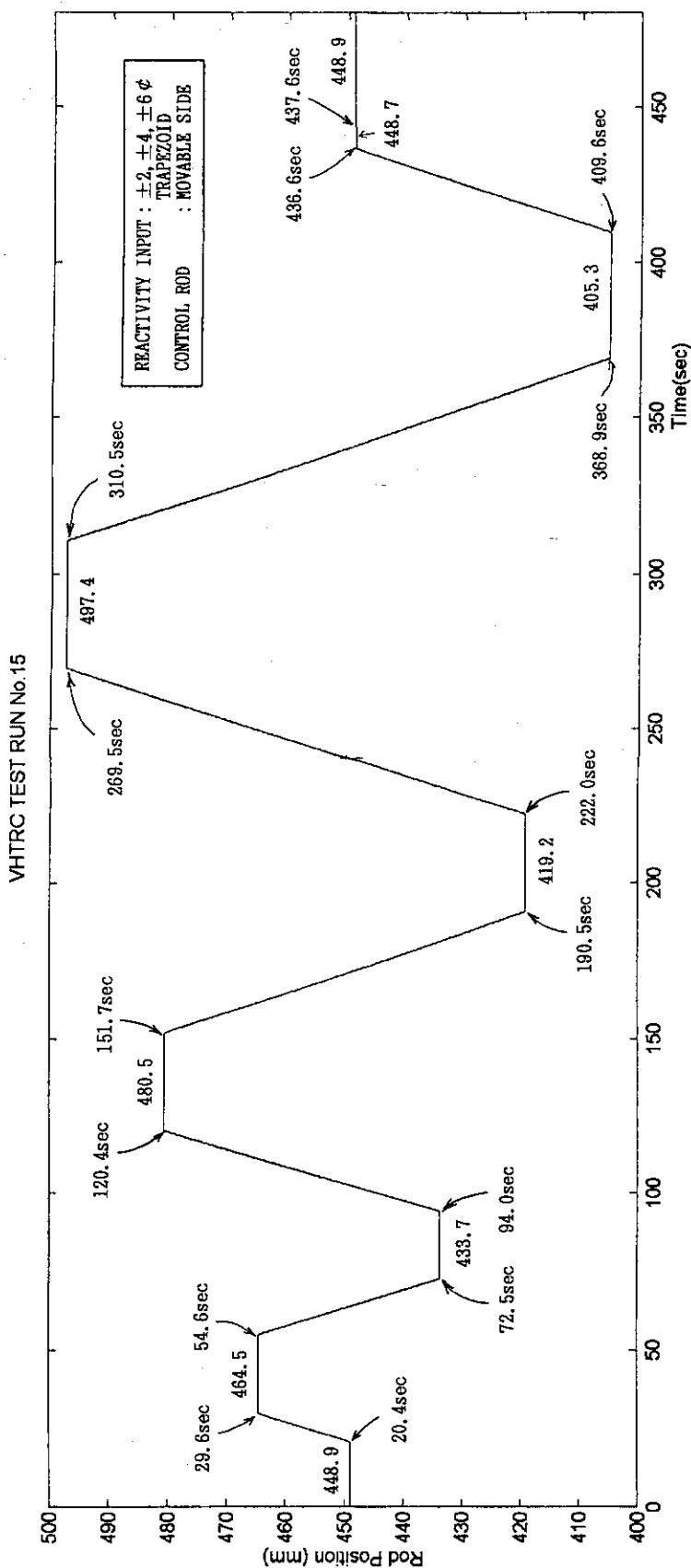


第4.47図 制御棒位置信号トランジエント部分の拡大図（実験No13）

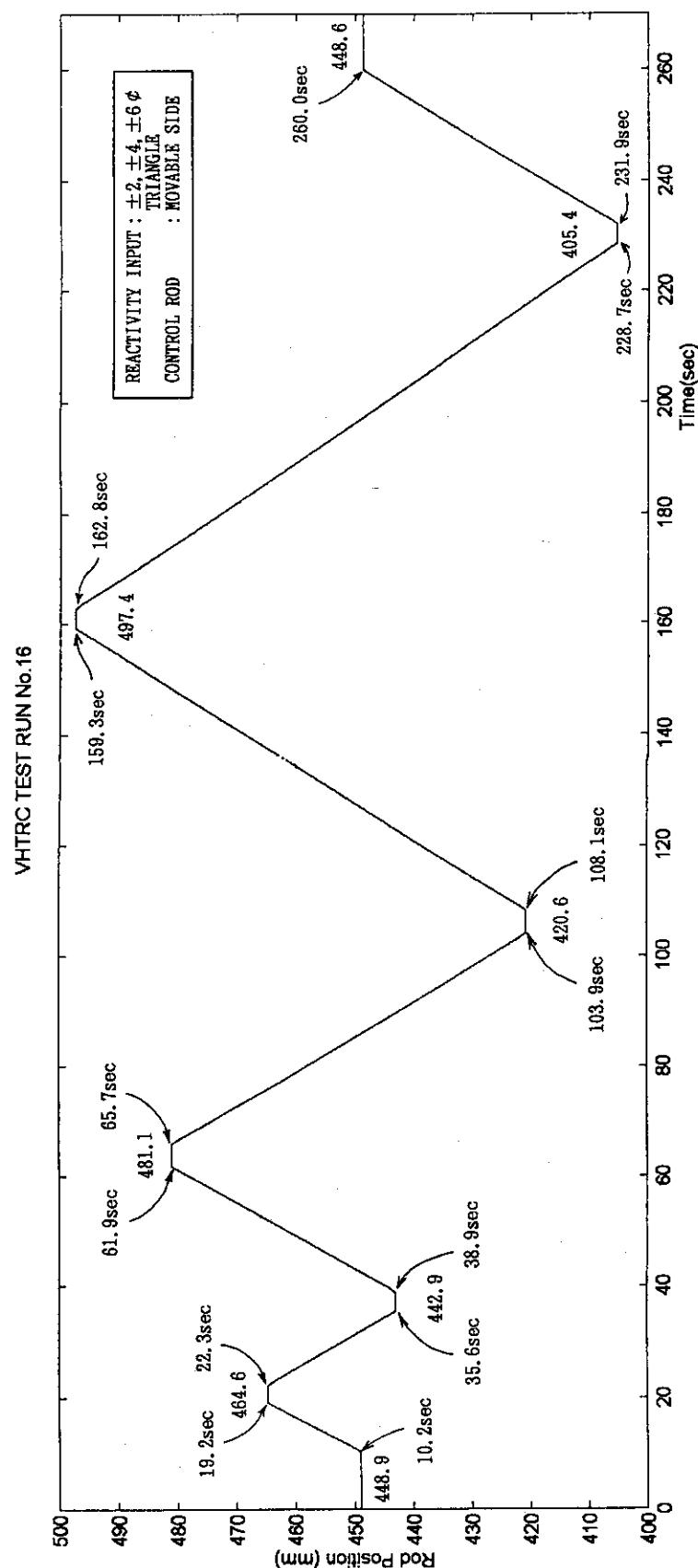
VHTRC TEST RUN No.14



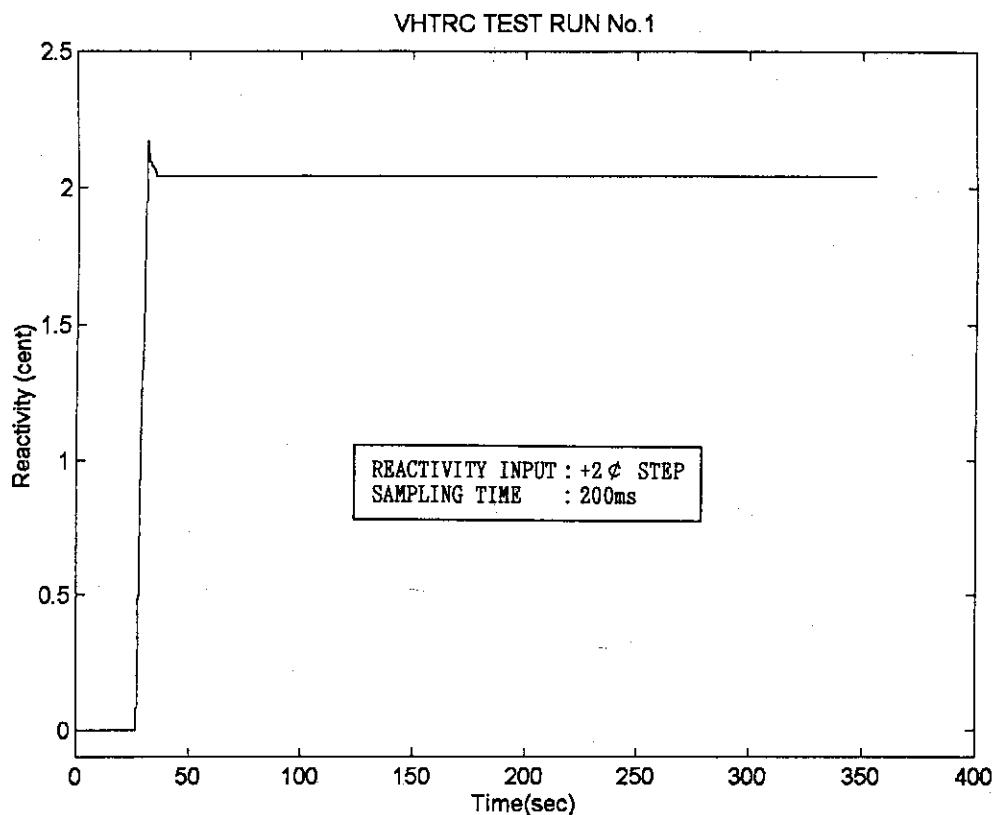
第4.48図 制御棒位置信号トランジエント部分の拡大図（実験No14）



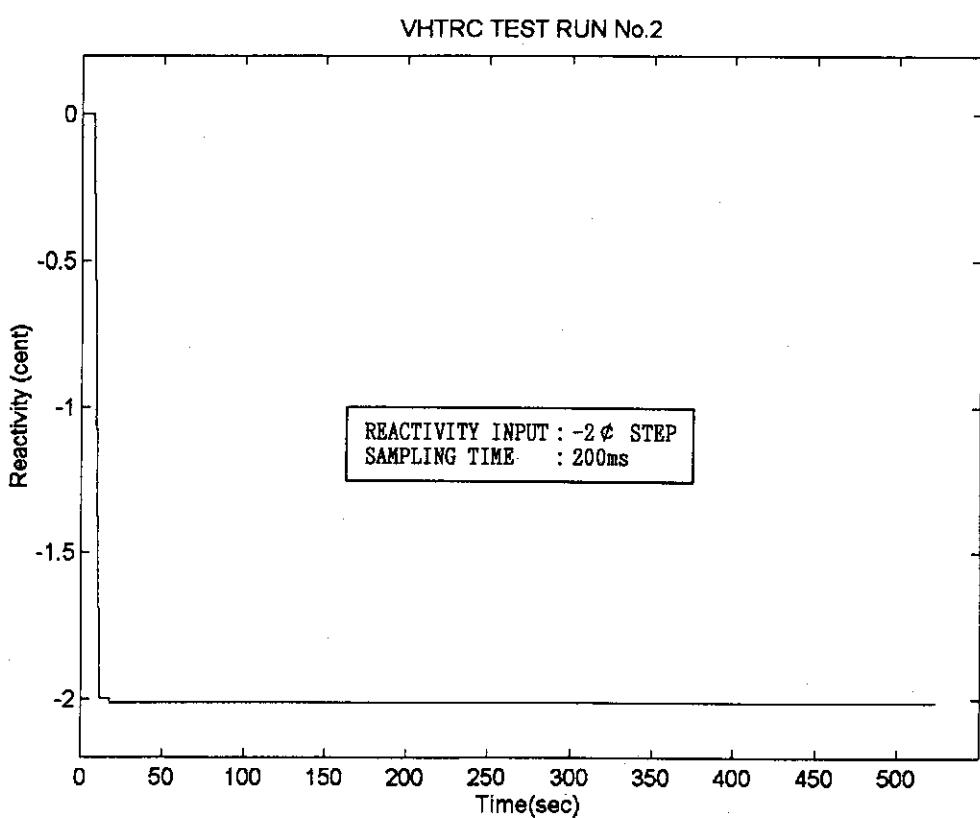
第4.49図 制御棒位置信号ランジメント部分の拡大図（実験No.15）



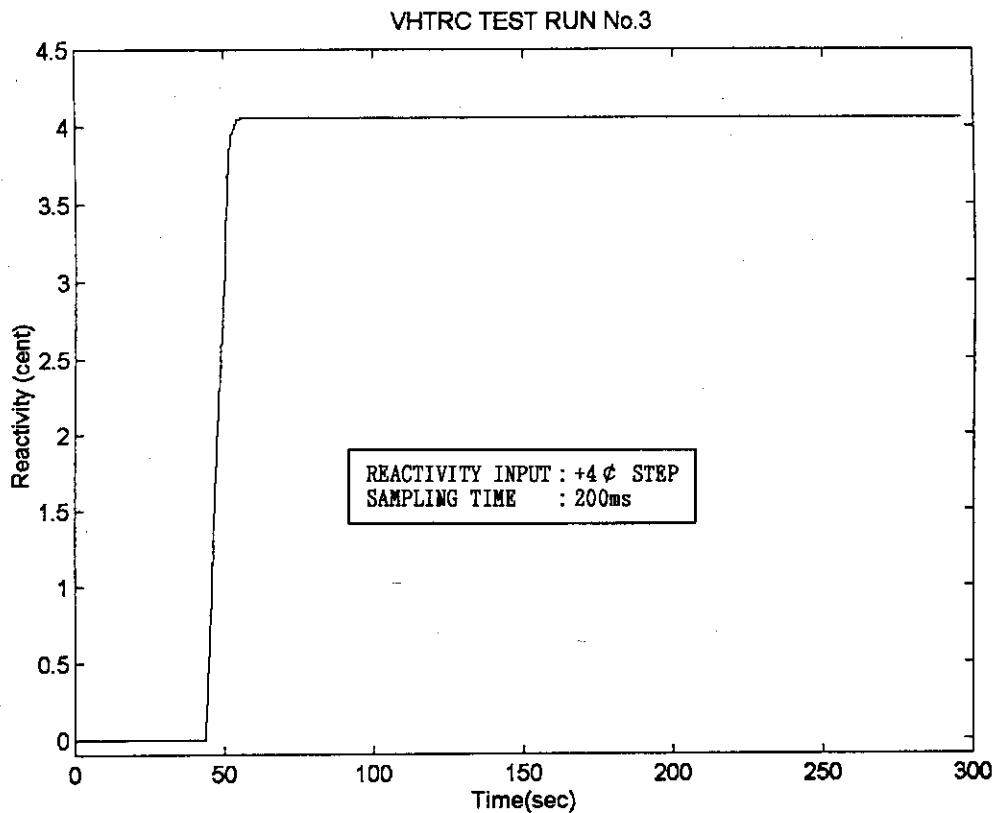
第4.50図 制御棒位置信号トランジエント部分の拡大図（実験No.16）



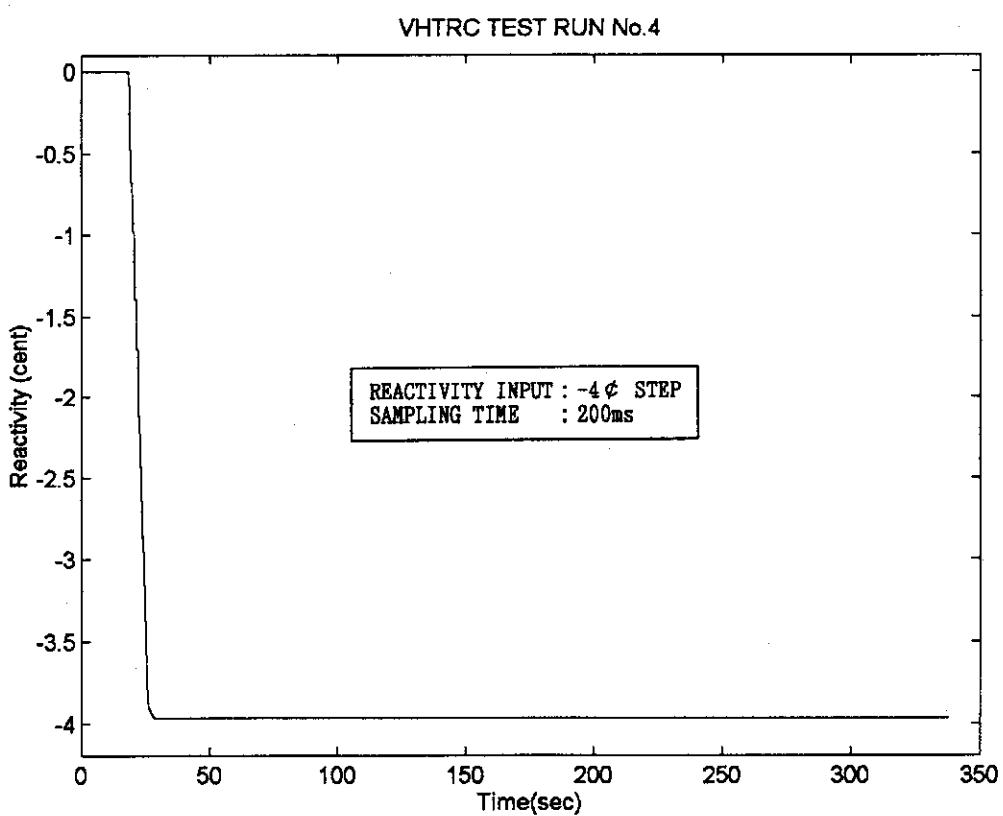
第4.51図 反応度変化量(実験No.1)



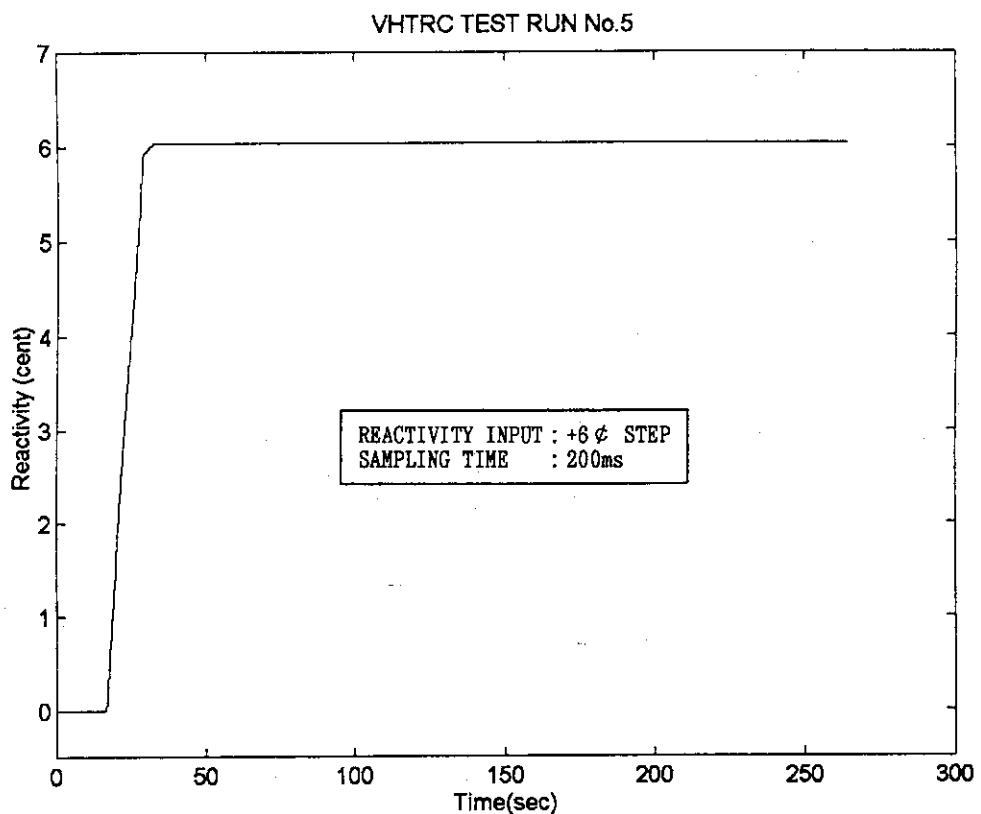
第4.52図 反応度変化量(実験No.2)



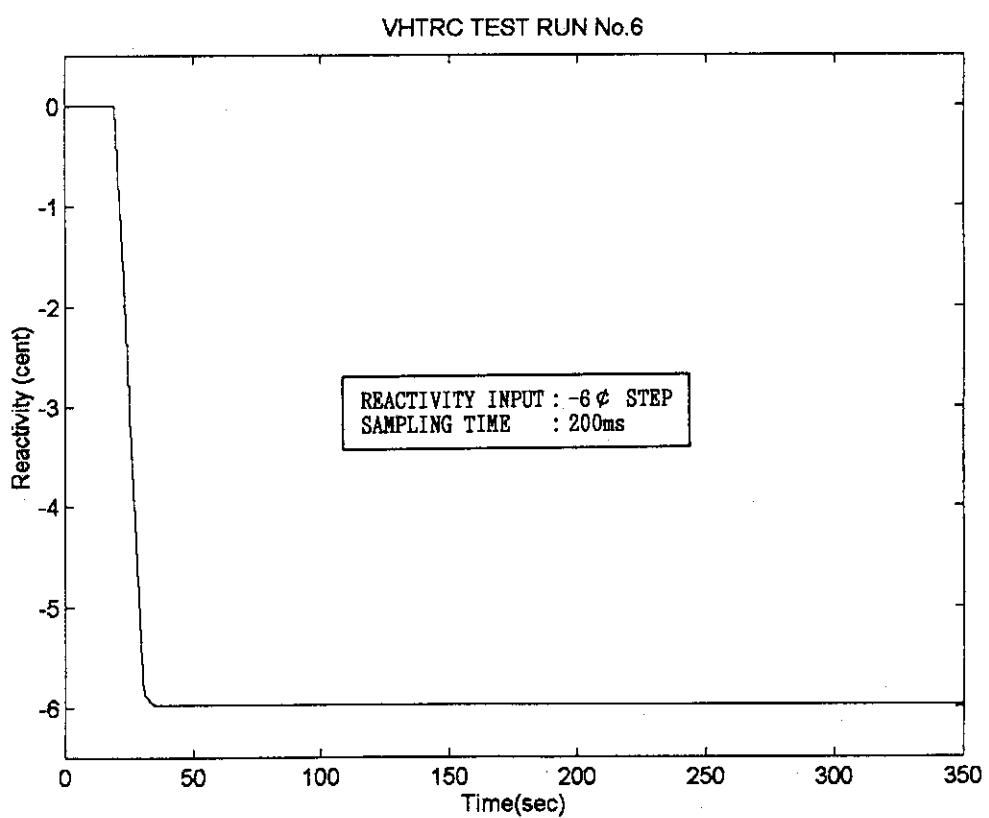
第4.53図 反応度変化量(実験No.3)



第4.54図 反応度変化量(実験No.4)

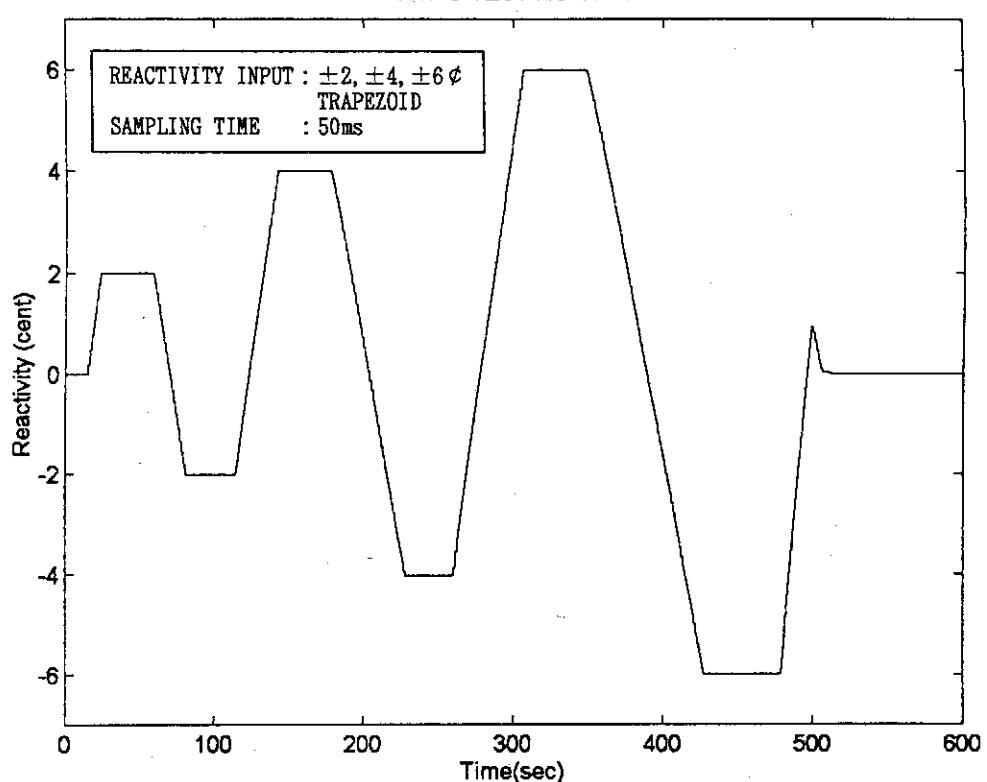


第4.55図 反応度変化量(実験No.5)



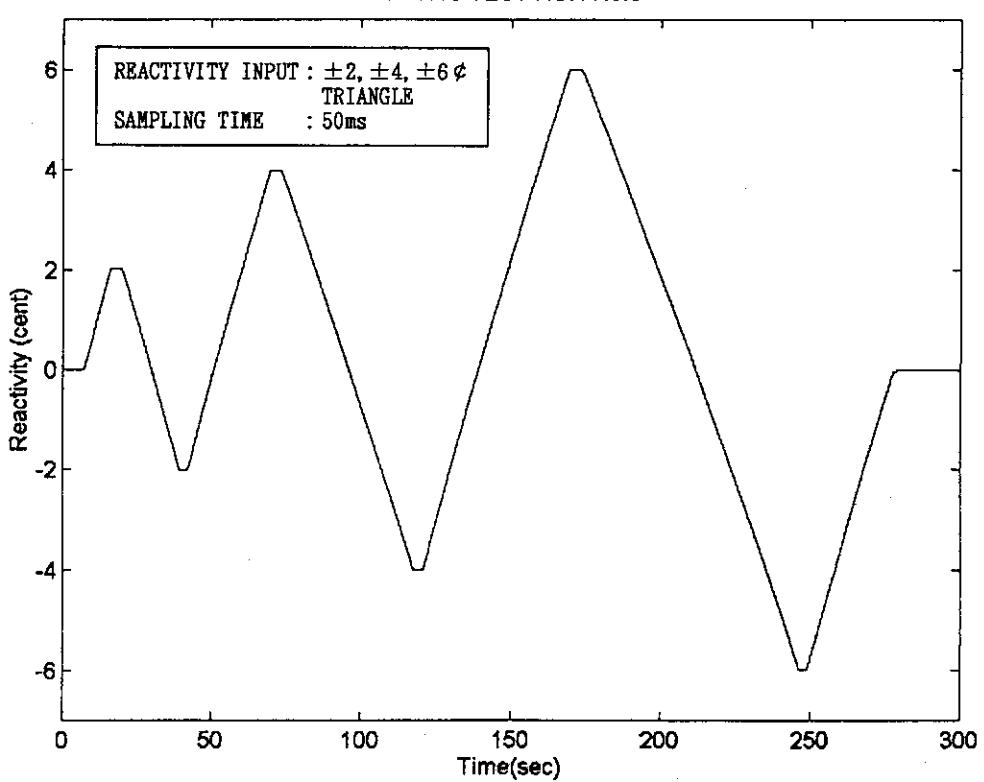
第4.56図 反応度変化量(実験No.6)

VHTRC TEST RUN No.7



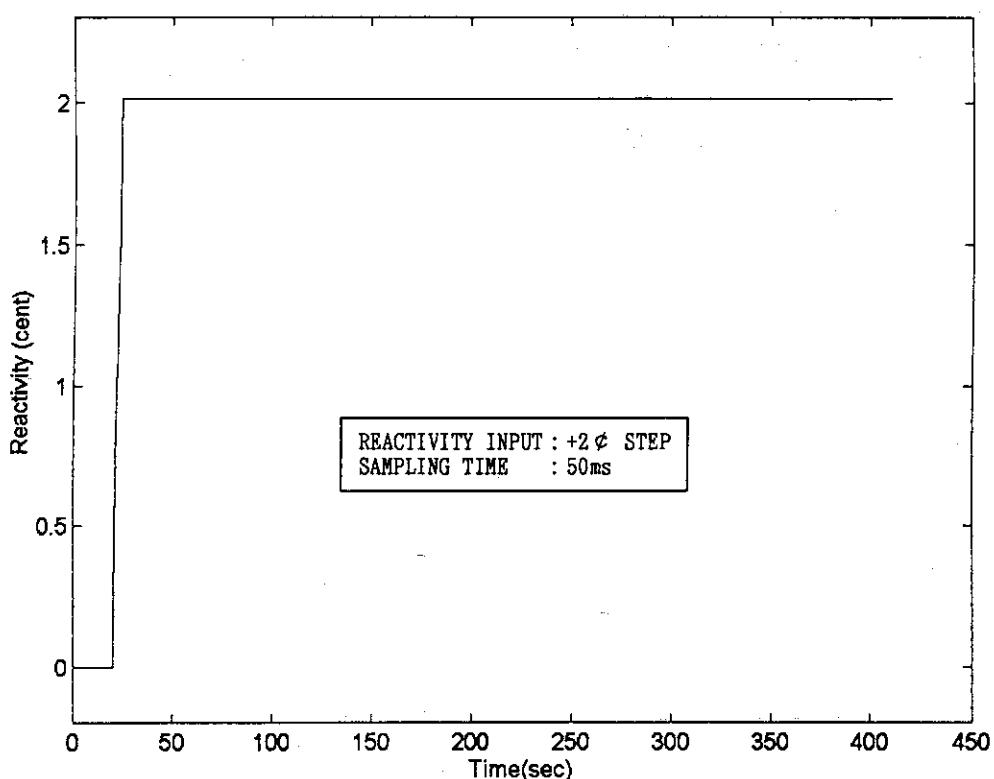
第4.57図 反応度変化量(実験No.7)

VHTRC TEST RUN No.8



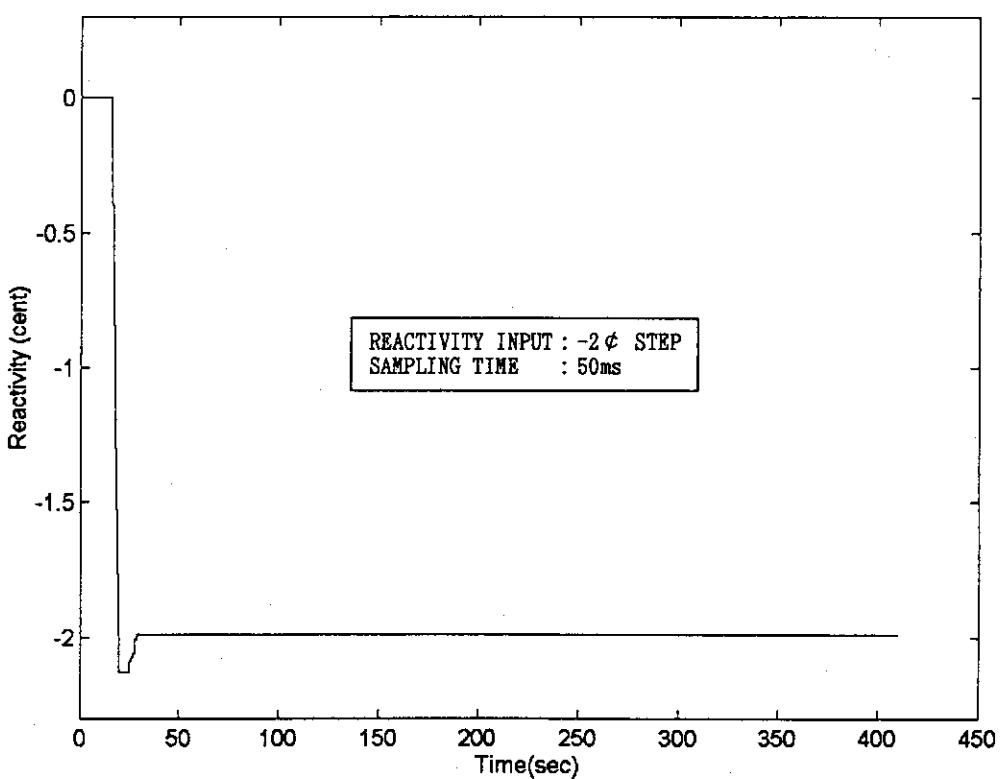
第4.58図 反応度変化量(実験No.8)

VHTRC TEST RUN No.9

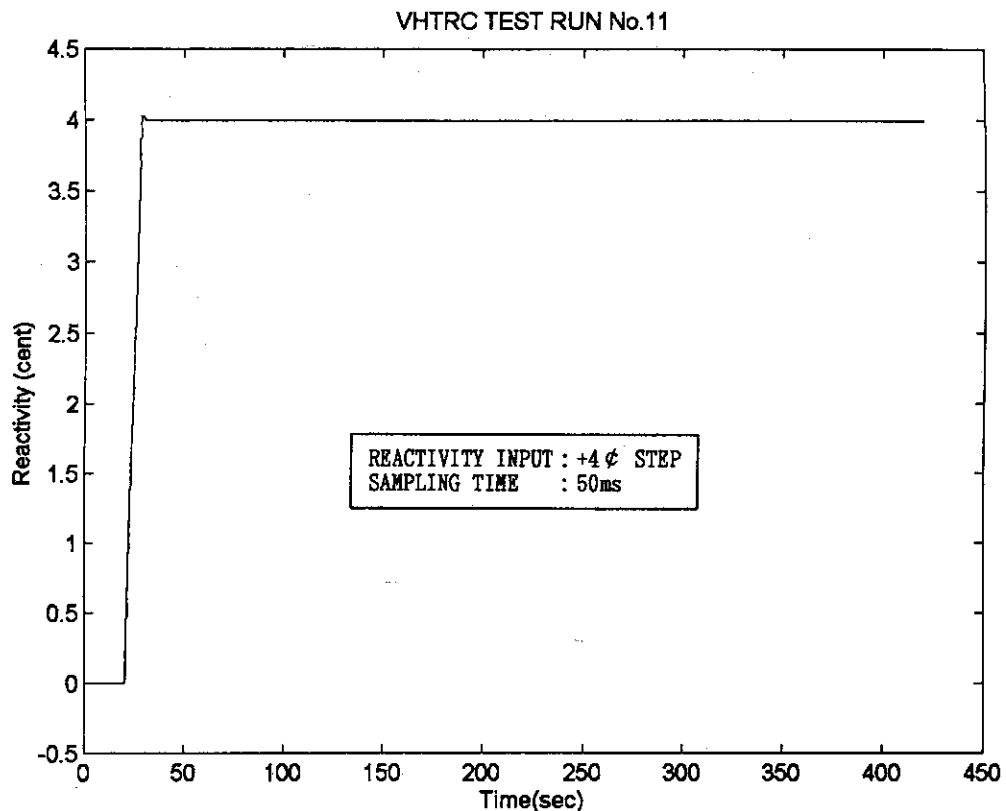


第4.59図 反応度変化量(実験No.9)

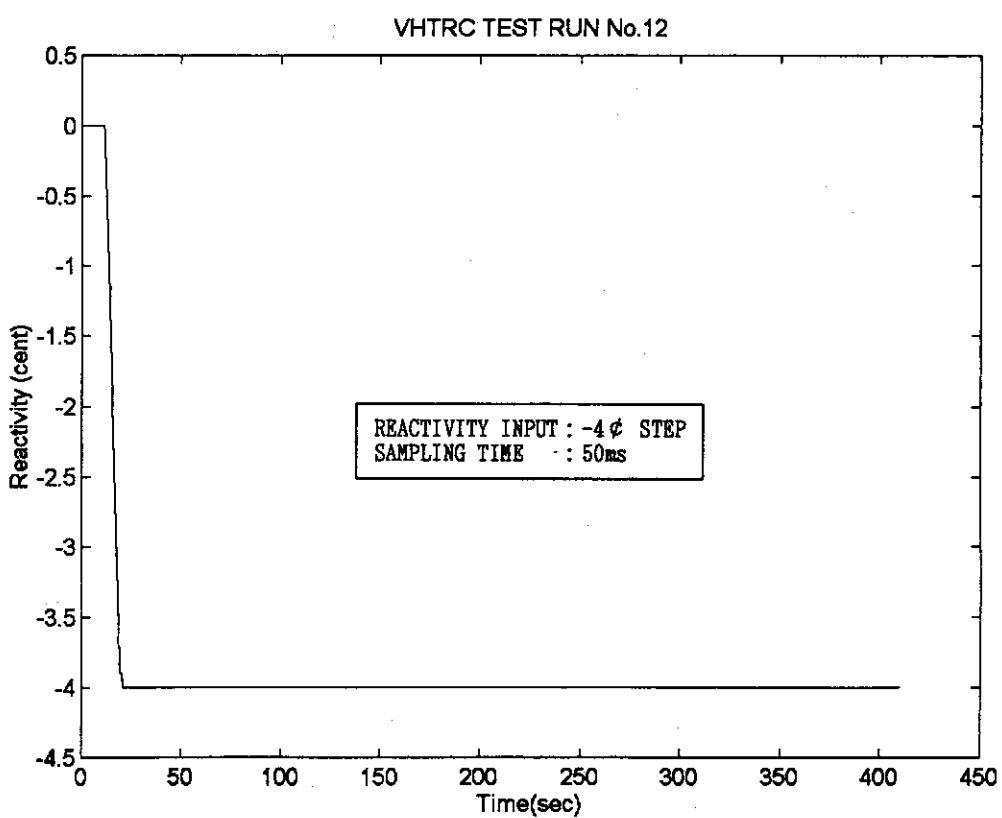
VHTRC TEST RUN No.10



第4.60図 反応度変化量(実験No.10)

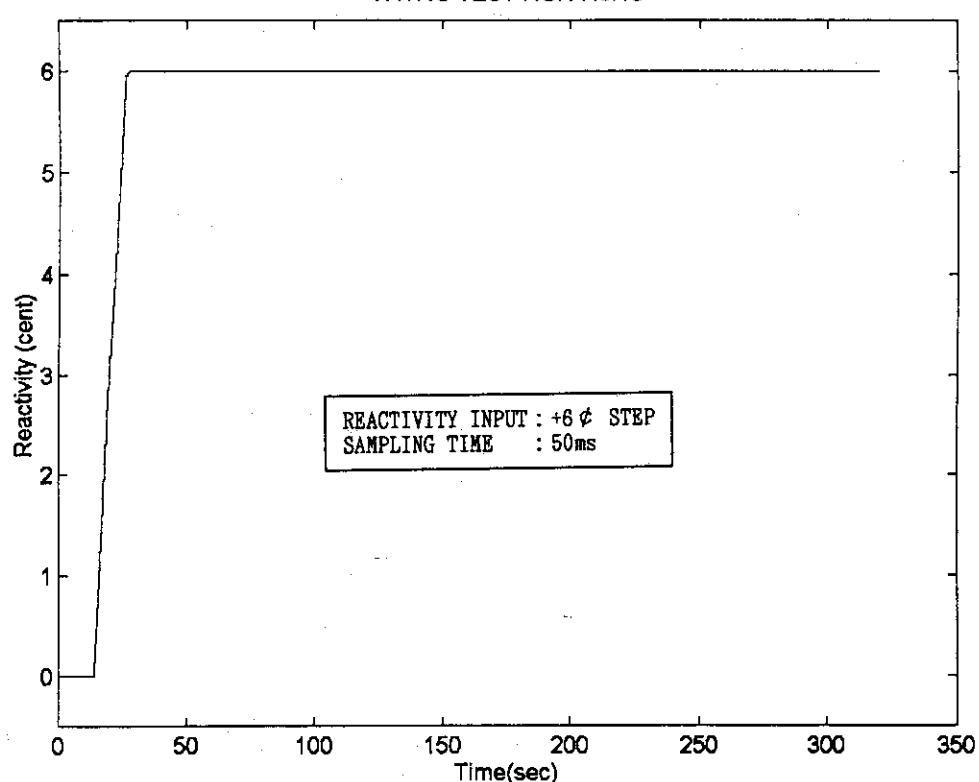


第4.61図 反応度変化量(実験No.11)



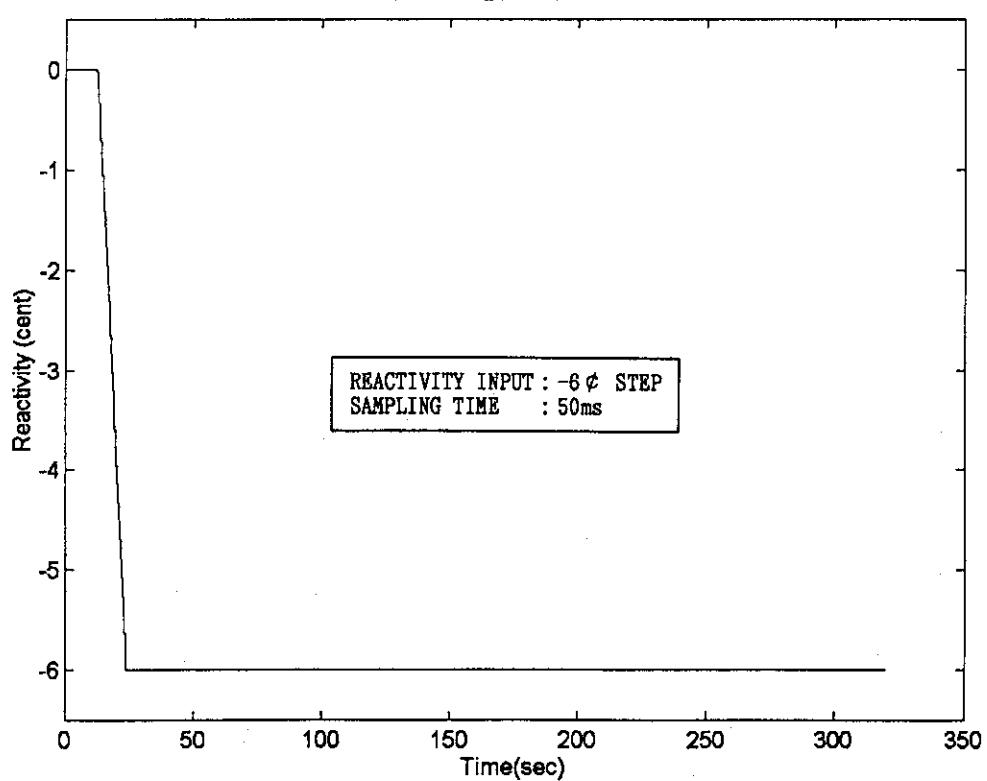
第4.62図 反応度変化量(実験No.12)

VHTRC TEST RUN No.13



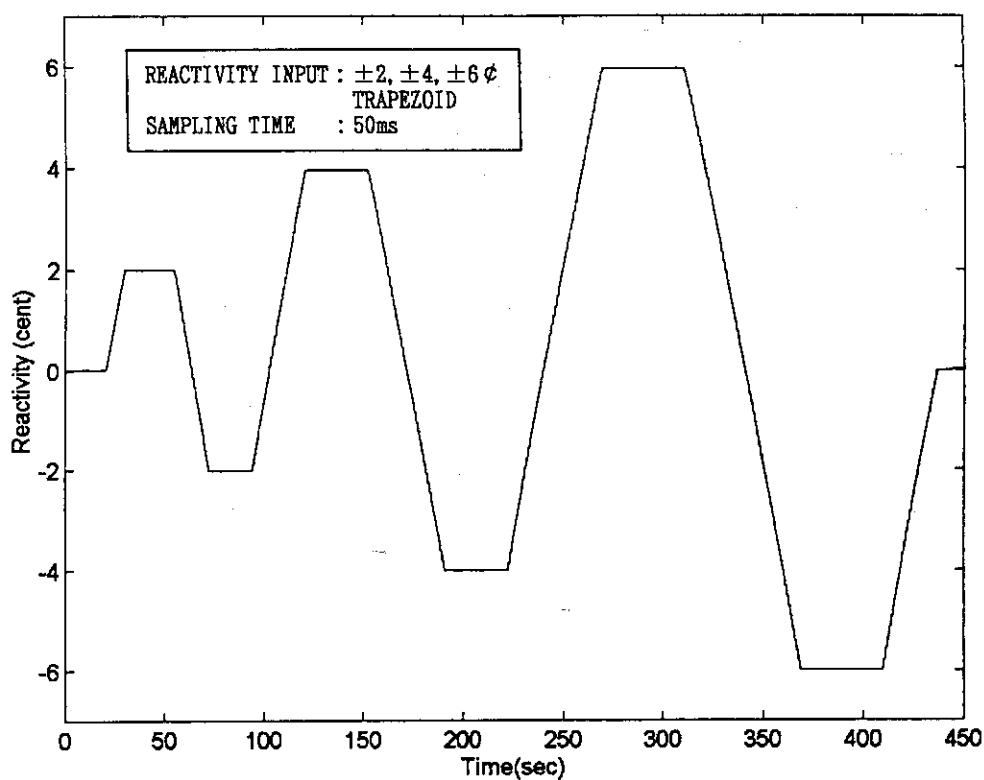
第4.63図 反応度変化量(実験No.13)

VHTRC TEST RUN No.14



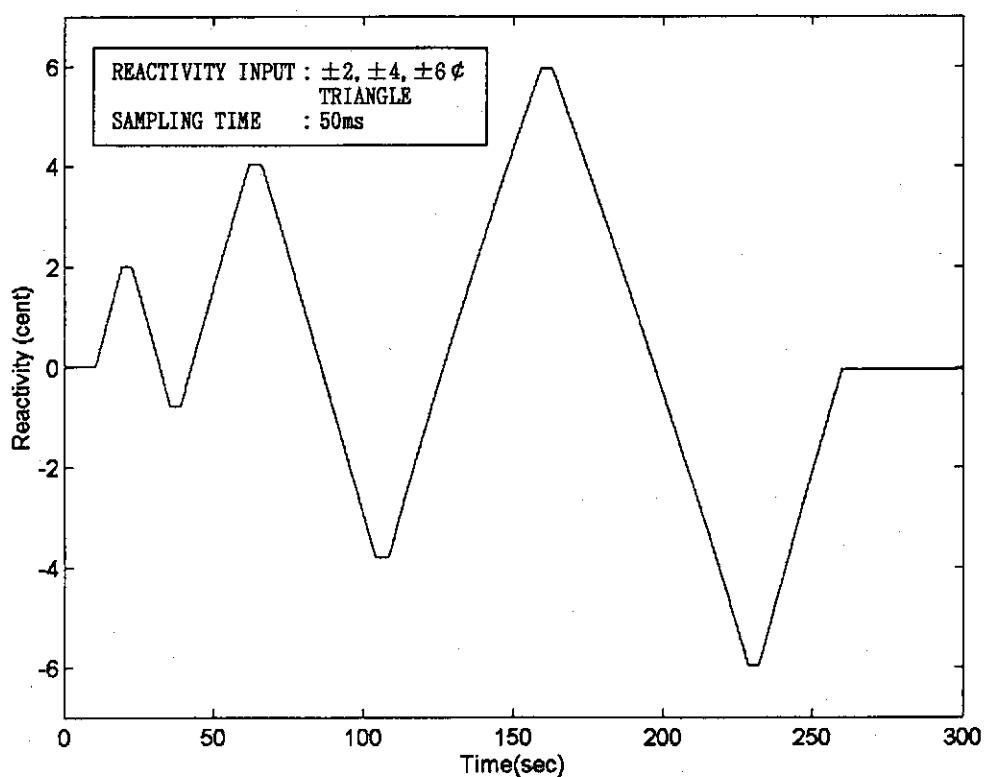
第4.64図 反応度変化量(実験No.14)

VHTRC TEST RUN No.15



第4.65図 反応度変化量(実験No.15)

VHTRC TEST RUN No.16



第4.66図 反応度変化量(実験No.16)

5. 考 察

今回のV H T R Cにおける実験を通して種々の知見を得た。その主なものを次にまとめる。

(1) 収録したリニアN炉出力信号系の中で、検出信号レベルの変化幅が大きく、挿入した反応度に対して顕著に応答している核計装チャンネルはCH1であり、この信号を反応度推定実験用に利用するのが最適と思われる。第5.1図にCH1とCH2のパワースペクトル密度を求めた結果を示す。同図は実験No.2のデータについてデータ収録装置のFFT解析機能を利用して出力したものであるが、周波数領域全般にわたりCH1信号の方がCH2信号に比べ約10db程度レベルが高くなっている。この傾向は全実験No.に共通しており、同一測定レンジにおいてはCH1が核計装チャンネルの中で最も反応度変化に対する感度が高いチャンネルと言える。従ってCH1の信号を反応度推定実験用に利用することとした。

(2) リニアN炉出力信号に含まれる比較的高い周波数領域の信号は、炉出力の上昇に伴って核計装装置の測定レンジを順次切換えていくと、その信号レベルは炉出力の低周波領域の信号に比し相対的に小さくなっていく。しかし、その高い周波数領域の信号のRMS値に着目してみると、今回の実験では0.1W(10^{-8} レンジ)から1W(10^{-7} レンジ)に炉出力を変更した際の、高い周波数領域の信号のRMS値は入力側換算で約3倍程度に上昇している。この高い周波数領域の信号は、V H T R Cの核的なゆらぎ信号とケーブルに誘導するノイズや検出系の内部から発生するノイズ等が重畠したものと考えられるが、その分離分析は今回の実験ではできなかった。第5.2図(1)にCH1の0.3W(10^{-8} レンジ)における高い周波数領域の信号を第5.2図(2)に3W(10^{-7} レンジ)における高い周波数領域の信号を示す。これらの高い周波数領域の信号はV H T R Cがそれぞれの出力レベルにおいて臨界状態になっている場合の直流分を差し引いたいわゆる炉雑音信号である。

第5.1表に各実験No.における高い周波数領域の信号のRMS値を示す。これはV H T R Cが臨界状態になっている場合の、10あるいは15秒間における炉雑音信号のRMS値を求めたものである。同表内の実験No.9~16の値はレンジ・ゲインの補正を行った後の値を示している。

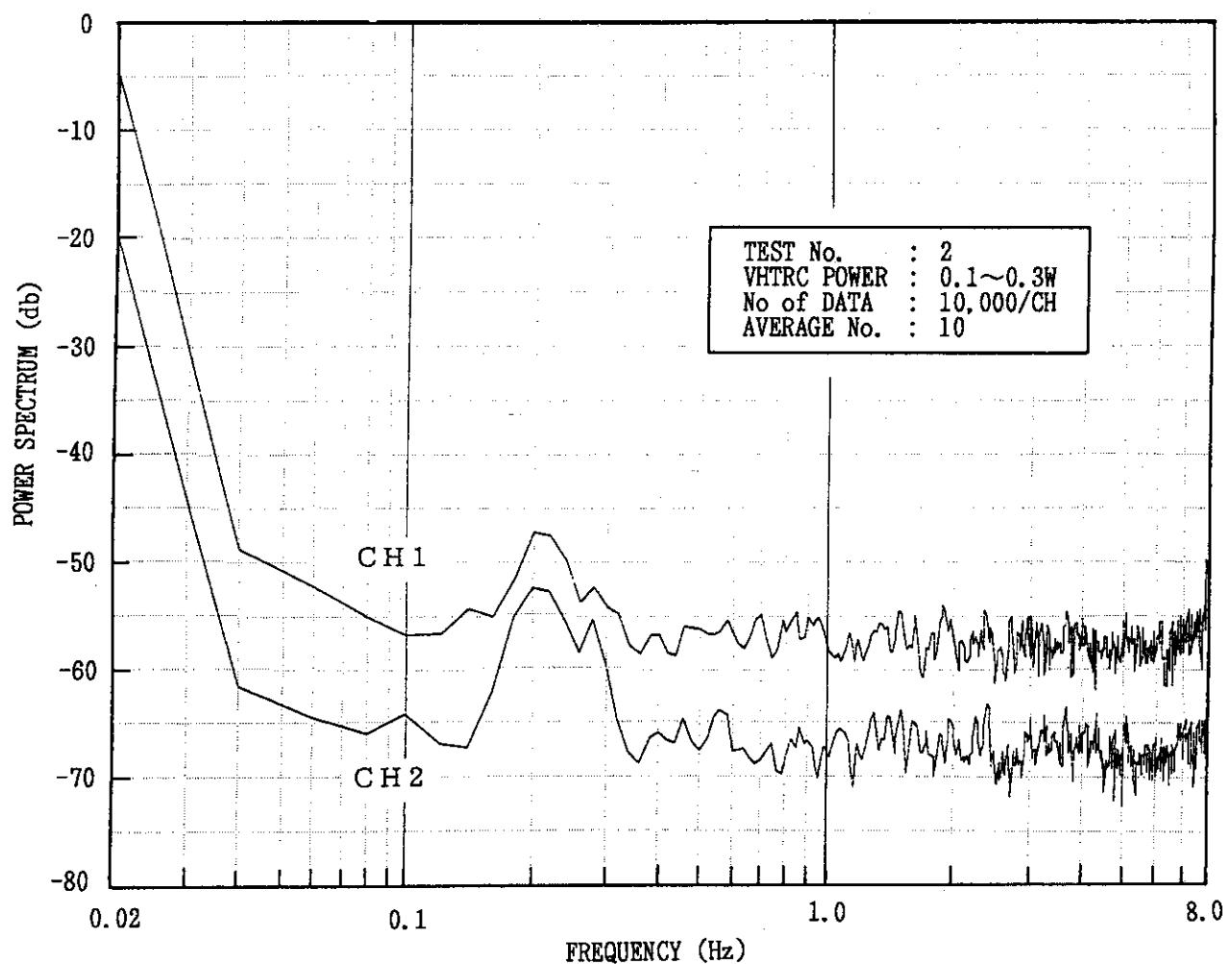
(3) 今回の実験の目的には直接入っていないが、炉出力信号のパワースペクトルを求めたところ、ほぼ全実験No.を通して、0.2~0.3Hz付近に小さなスペクトルのピークがあることが認められた。このピークを生じさせるゆらぎの原因は現在のところ不明である(第5.1図参照)。

- (4) 今回の実験では信号のサンプリング周期を 50 msecにしたが、 H^∞ 反応度推定方式を詳細に評価したり、炉出力信号のノイズ解析をするためにはもう少し高速でサンプリングしておく必要があったものと考えられる。また収録したデータを多方面に活用するためにも、臨界状態における炉出力信号の計測をもっと長い時間にわたって行うべきであったと思われる。しかし、これを実現するためには、データ収録装置のメモリ容量(最大4 Mワード)の関係から、計測する信号の種類を減らさなければならない制約もある。
- (5) 希望する反応度変化を正確にかつ容易に実現したり、いろいろな検出信号を比較的容易に利用できるなど、今回の H^∞ 反応度推定実験の対象として、V H T R C は最適な原子力施設であった。

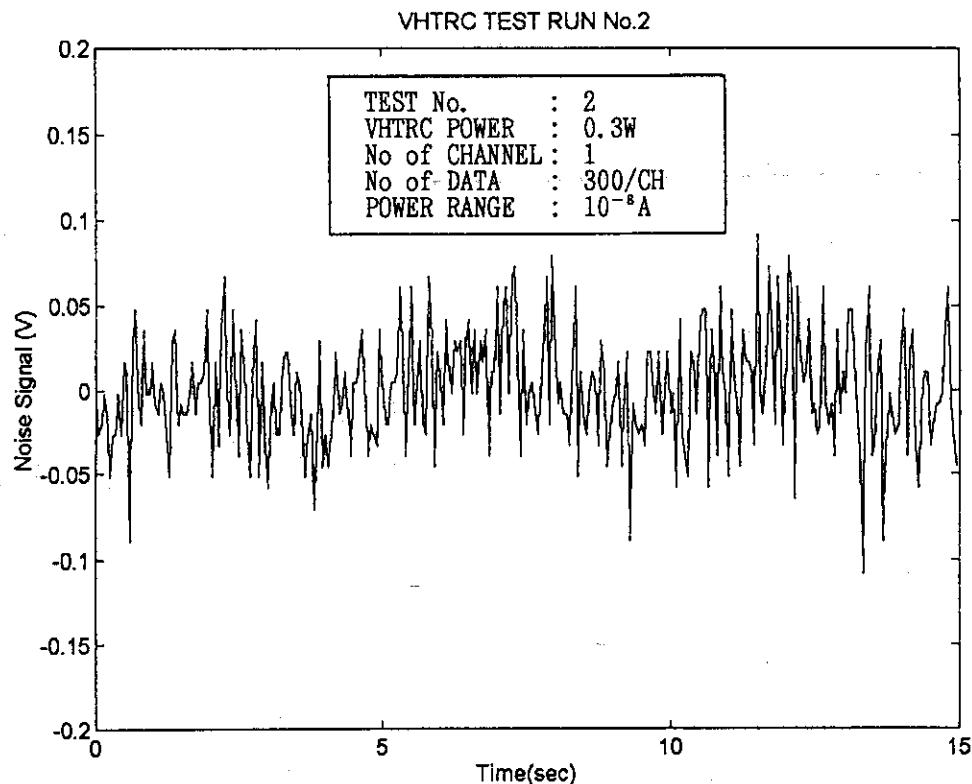
第5.1表 VHTRC炉雑音信号のRMS値

実験No.	炉雑音信号のRMS値(mV)	炉出力(W)	測定レンジ(A)	計測時間(sec)
1	32.60	0.1	10 ⁻⁸	15
2	33.75	0.3	〃	〃
3	34.46	0.1	〃	〃
4	36.89	0.3	〃	〃
5	32.56	0.1	〃	〃
6	36.06	0.5	〃	〃
7	28.13	0.35	〃	〃
8	35.98	0.35	〃	〃
9	89.63	1.0	10 ⁻⁷	〃
10	105.46	3.0	〃	〃
11	92.20	1.0	〃	〃
12	110.96	5.0	〃	10
13	90.65	1.0	〃	15
14	113.05	5.0	〃	10
15	100.47	2.0	〃	15
16	100.66	2.0	〃	10

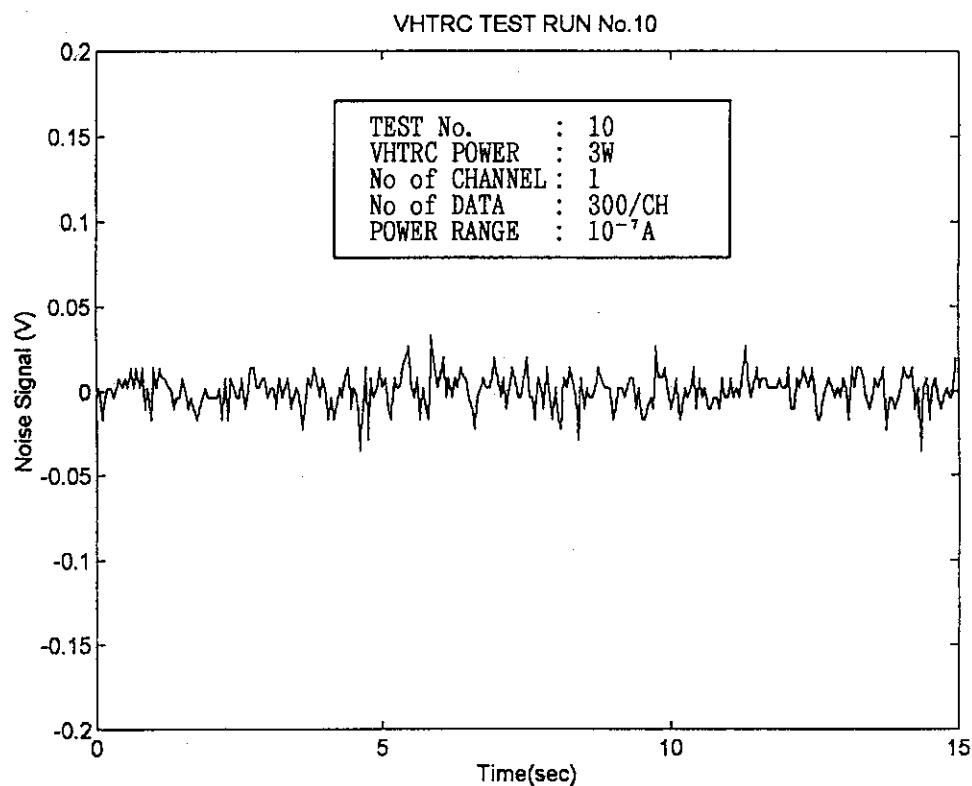
注) 10⁻⁷レンジにおけるRMS値は10⁻⁸レンジにおけるRMS値と比較するために、
10⁻⁷レンジにおける測定値を10倍して求めた値である。



第5.1図 VHTRC炉出力信号のパワー・スペクトル密度



(1) 0.3 W出力時



(2) 3 W出力時

第5.2図 VHTRCの炉雑音信号

6. おわりに

今回のVHTRCにおけるH ∞ 反応度推定実験は極めてスムーズに、予定したすべてのデータ収録作業を実施することができた。今回の実験を実施するにあたっては、VHTRCの計装制御系に改造等の手を加えること無しに、出来るだけ多くの計測信号を収録するように心掛けた。幸いにVHTRCにはデータロガーが早い時期から導入されており、計測信号の取り出しは容易に実現できた。VHTRC建屋におけるPhase-1の実験は1995年9月21日と22日の両日に渡り実施した。当初、すべての項目の実験実施にはかなりの時間を必要とするのではないかと思われたが、VHTRCの運転が極めてスムーズに短時間で各運転条件を確保することができたため、予想よりもはるかに短い時間で実験を終了することができた。むしろ、その後のデータ処理作業の方に多くの時間を費やした。今回の実験で収録したデータが比較的大容量のファイルになるため、コード変換などのデータ処理作業に多くの時間を必要とした。しかし、今回のデータ収録に関する反省点としては、さらに膨大なデータファイルになってしまふが、データのサンプリングタイムを現在の50 msecからさらに10 msec程度に短くしておけば良かったと思われることである。これを実現するためには、データ処理をする計算機のメモリの容量や処理プログラムについての検討が必要となるが、H ∞ 反応度推定方式を詳細に評価する段階で、多角的な評価が可能になるという点で惜しまれることであった。もしも再度測定実験をする機会があれば、データ収録に要する時間の目安も今回の実験で把握することができたので、条件を変えて実施してみたいと考えている。また、今回はデータ処理をパソコンとSPARC2ワークステーションの2つに分けて実施したが、大容量のデータファイルを多数回に渡り両計算機間を転送させる作業は極めて非能率的であった。今後のデータ処理作業に関する改善すべき点の1つであろう。

今回のデータ処理用に新たに作成した各種のプログラムはすべてC言語で作成してあるが、できるだけ処理項目毎にサブルーチン化してあるため分りやすく、他のプログラムにも応用できる要素を数多く含んでいるので、今後の応用が期待される。

今回の実験において収録したデータの詳細な解析の結果や、取得した中性子束信号をもとにH ∞ 制御理論による反応度推定を行った結果については統報にまとめる予定である。

参考文献

- 1) 篠原慶邦他：高速実験炉の反応度最適推定方式の研究，(1973)
- 2) 安田秀志他：V H T R C（高温ガス炉臨界実験装置）の建設，(1987)
- 3) 藤井義雄他：積分型広域反応度計の試設計，(1974)
- 4) 藤井義雄他：積分型広域反応度計の開発(I)，(1977)
- 5) 藤井義雄他：積分型広域反応度計の開発(II)，(1978)
- 6) 日置電機株式会社：8825メモリハイコーダ取扱説明書，(1994)
- 7) 日置電機株式会社：9536 UTILITY DISK取扱説明書，(1994)
- 8) サイバネットシステム株式会社：MATLABユーザーズガイド，(1994)

付録1 ディジタルデータ収録装置の仕様^{①)}

1.1 概要

今回の実験に使用したディジタルデータ収録装置は日置電機製8825型メモリハイコーダで、多くの優れた性能を備えた可搬型の収録装置である。その主な性能について次に示すが、取扱方法やもっと詳しい仕様内容については、同装置のマニュアルを参照されたい。ここでは同装置の主要な仕様、今回の実験で利用した機能等について重点的に記載する。

1.2 ディジタルデータ収録装置の仕様

8825型ディジタルデータ収録装置(商品名:メモリハイコーダ)は信号の波形観測と同時にメモリへのデータ記録ができる、汎用的な波形記録計である。その主な仕様を次に述べる。

1.2.1 本体およびアナログ入力ユニットの仕様

- (1) 測定機能 : レコーダ (実時間記録)
メモリレコーダ (高速記録)
X-Yレコーダ (連続X-Y記録)
FFT (FFT解析)
- (2) 入力チャンネル数 : アナログ16CH (フローティング入力)
2CHずつのプラグイン入力ユニット方式
ロジック8CH (4点/CH)
- (3) メモリ容量 : 最大4Mワード
- (4) 最高データ収納容量 : 4Mデータ
2Mデータ/CH (1~2CH使用時)
1Mデータ/CH (3~4CH使用時)
500Kデータ/CH (5~8CH使用時)
200Kデータ/CH (9~16CH使用時)
- (5) 最高サンプリング速度 : 200kSample/sec(メモリレコーダファンクション選択時)
但しレコーダファンクション選択時は最高560~120 μsec(使用CH数によって変る)
- (6) A/D変換分解能 : 12 ビット
- (7) サンプリング・A/D変換タイミング : 全チャンネル同時
- (8) アナログ測定レンジ : 5 mV/DIV~20 V/DIV (12 レンジ)
- (9) 外部制御端子 : 外部トリガ入力、トリガ出力、GO-NG出力、外部スタート/ストップ、プリント入力
- (10) 時計機能 : オートカレンダ、24時間計
- (11) 時間軸精度 : ±0.01% (目盛と時間との相対誤差を示す)
- (12) 外形寸法 : 約230H×385W×397Dmm
- (13) 重量 : 約10.7kg (本体のみ)

1.2.2 記録部の仕様

- (1) 記録方式 : サーマルラインヘッドによる感熱記録方式
- (2) 記録紙 : 264mm×30m ロール型感熱記録紙

(3) 記録幅	: 全記録幅 256mm(2048ドット)
	波形部 240mmfs
(4) 記録速度	: 最大約25mm/sec
(5) 紙送り寸法確度	: ±3% (25°C, 60%RH)

1.2.3 表示部の仕様

(1) 表示体	: 10インチプラズマディスプレイ(4階調表示) 640×480ドット
(2) 表示分解能	: 1 DIV=20 ドット(縦)×20 ドット(横) ・メモリレコーダ、レコーダ時 波形部 20DIV fs×25DIV fs(ノーマル) 24DIV fs×25DIV fs(ワイド) 文字部 80文字×30行(8×16ドット) ・X-Yレコーダ時 波形部 20DIV fs×20DIV fs(ノーマル) 24DIV fs×24DIV fs(ワイド) 文字部 80文字×30行(8×16ドット)
(3) ドットピッチ	: 0.31mm×0.31mm

1.2.4 外部記憶部の仕様

(1) 記憶装置	: 3.5インチ フロピーディスクドライブ
(2) 記憶容量	: 1.2Mバイト(2HD)(PC9801用) 720kバイト(2DD)(IBM-PC用)
(3) データフォーマット	MS-DOS対応
(4) 記憶内容	: 設定条件、測定データ、波形測定エリア 測定データはA・Bカーソル間の部分データのセーブが可能

1.2.5 トリガ部の仕様

(1) トリガ方式	: デジタル比較方式
(2) トリガモード	: シングル、リピート、オート ただしレコーダモード時のオートはなし
(3) トリガソース	: 各チャンネル毎にトリガ条件を設定可能
(4) その他の機能	: プリトリガ機能(メモリレコーダモード) パルントリガ機能(ロジック入力)

1.3 デジタルデータ収録装置の機能

デジタルデータ収録装置は大別すると①メモリレコーダ・ファンクション②レコーダ・ファンクション③FFT(高速フーリエ変換)の3つの測定モード(機能)を備えている。メモリレコーダ・ファンクションとレコーダファンクションはともに波形を表示するとともに、サンプリングしたデータをメモリ内に順次ストアしていく。この2つのモードの大きな相違点は次の3点に集約できる。

- ① メモリレコーダ・ファンクション・モードにおけるサンプリングタイムは時間軸の設定値によって変化するが、レコーダ・ファンクション・モードにおけるそれは、時間軸の数値にかかわらず一定である。ただし使用するチャンネル数によって変る。メモリレコーダ・ファンクション・モードはレコーダ・ファンクション・モードよりも高速サンプリングが可能である。
- ② メモリレコーダ・ファンクション・モードにおけるデータの記録長は、使用するチャンネル数によっても変るが、最高20,000DIVまで可能であるが、レコーダ・ファンクション・モードにおけるそれは最高1,000DIVであり、それを越えるデータを入力した場合には順次データが更新され最後の1,000DIVのデータがメモリに保存される。
- ③ メモリレコーダ・ファンクション・モードにおける時間軸の設定可能範囲は、レコーダ・ファンクション・モードよりも高速領域である。

次に各測定モードの仕様を示す。

1.3.1 メモリレコーダ・ファンクション

- (1) 時間軸 : 500 μsec/DIV, 1, 2, 5, 10, 20, 50, 100, 500msec/DIV
1, 2, 5, 10, 20sec/DIV 1, 2, 5min/DIV
- (2) 時間軸分解能 : 100ポイント/DIV
- (3) サンプリング周期 : 時間軸の1/100
- (4) 記録長 : 20,000 DIV(1~2CH使用時)
10,000 DIV(3~4CH使用時)
5,000 DIV(5~8CH使用時)
2,000 DIV(9~16CH使用時)
1,000, 500, 200, 100, 50, 25 DIV
- (5) 表示フォーマット : シングル、デュアル、クワッド、オクト(プリントのみ)
ヘキサ(プリントのみ)、X-Yシングル、X-Yクワッド
- (6) 補間機能 : あり、ドット、ライン
- (7) 波形拡大・圧縮機能(時間軸倍率) : 10, 5, 2, 1, 1/2, 1/5, 1/10, 1/20, 1/50, 1/100, 1/200
1/500, 1/1000
(電圧軸倍率) : 16, 8, 4, 2, 1, 1/2
- (8) 波形スクロール : 上下左右方向にスクロール可能
- (9) プリント機能 : オートプリントON.OFF, マニュアルプリント
部分プリント可能
- (10) ロギング機能 : 測定データのデジタル値プリント
- (11) 演算処理機能 : 四則演算, 絶対値, 指数, 常用対数, 平方根, 移動平均,
微分(1次, 2次), 積分(1次, 2次), 時間軸方向の平行移動,
波形パラメータ演算(MAX値, MIN値, P-P値, 平均値
RMS値, 面積値, 周期, 周波数, 立ち上がり時間など)
- (12) アベレージング機能 : 加算平均, 指数化平均

1.3.2 レコーダ・ファンクション

- (1) 時間軸 : 200msec/DIV(ディスプレイのみ), 500msec/DIV
1, 2, 5, 10, 20sec/DIV 1, 2, 5, 10, 20min/DIV, 1h/DIV

(2) 時間軸分解能	: 100ポイント/DIV
(3) サンプリング周期	: 時間軸にかかわらず一定 120 μ sec/DIV (1~2CH使用時) 160 μ sec/DIV (3~4CH使用時) 280 μ sec/DIV (5~8CH使用時) 560 μ sec/DIV (9~16CH使用時)
(4) 記録長	: 25, 50, 100, 200, 500DIV CONT
(5) 表示フォーマット	: シングル、デュアル、クワッド、オクト(プリントのみ) ヘキサ(プリントのみ)
(6) 機間機能	: あり、ライン表示のみ
(7) 波形スクロール	: 上下左右方向にスクロール可能
(8) プリント機能	: オートプリントON/OFF, 再プリント可能
(9) ロギング機能	: 測定データのデジタル値プリント

1.3.3 FFTファンクション

FFTは時間軸波形を周波数成分に分解する計算方法の一つである。今回の実験に使用したデジタルデータ収録装置は、演算処理の方法として2つの処理モードを備えている。1つはメモリレコーダとして取り込んだ入力データについてFFT演算をする方法であり、もう1つはFFTファンクション・モードで新たにデータを取り込みながらFFT演算する方法である。いずれの方法も、演算結果をグラフ化しディスプレイに表示したり、プリントアウトすることができる。

(1) FFTチャンネルモード	: 1CH FFT, 2CH FFT
(2) 周波数レンジ設定	: 133mHz~80kHz
(3) ダイナミックレンジ	: 72db
(4) サンプリング点数	: 1000 点
(5) 周波数分解能	: 1/400
(6) アンチエイリアジングフィルタ	: 周波数レンジに連動してカットオフ周波数を自動設定, ON/OFF設定可能
(7) 解析チャンネル設定	: 1CH FFT, 2CH FFTとも任意のチャンネル設定可能
(8) FFT解析モード	: ストレージ波形, リニア・スペクトラム, RMS・スペクトラム, パワース・ペクトラム, 自己相関関数, 頻度分布, 伝達関数, クロスパワー・スペクトラム, 相互相関関数, インパルス応答, コヒーレンス関数, オクターブ分析
(9) 表示フォーマット	: シングル, ディアル, ナイキスト表示, アレイ表示
(10) ウィンドウ	: 方形窓関数, ハニーベルト窓関数, エクスponenシャル窓関数
(11) アベレージング回数	: 2~4096
(12) 演算データの設定	: NEW DATA, FROM MEMORY
(13) 表示スケール設定	: AUTO, MANUAL

付録 2 制御棒位置データ変換プログラム

2.1 概要

VHTRCにおいて収録したデータファイルのフォーマットは時間軸データを含めたアナログデータチャンネル6CH+制御棒位置信号バイナリ32ビットの構成になっている。この変換プログラムはバイナリ32ビットを2つのフローティング・タイプの制御棒位置信号データに変換し、アナログデータ6CHの後にこの2つのデータを追加するものである。本プログラムはSPARC2ワークステーションを用いてコンパイルした。

2.2 プログラムの仕様

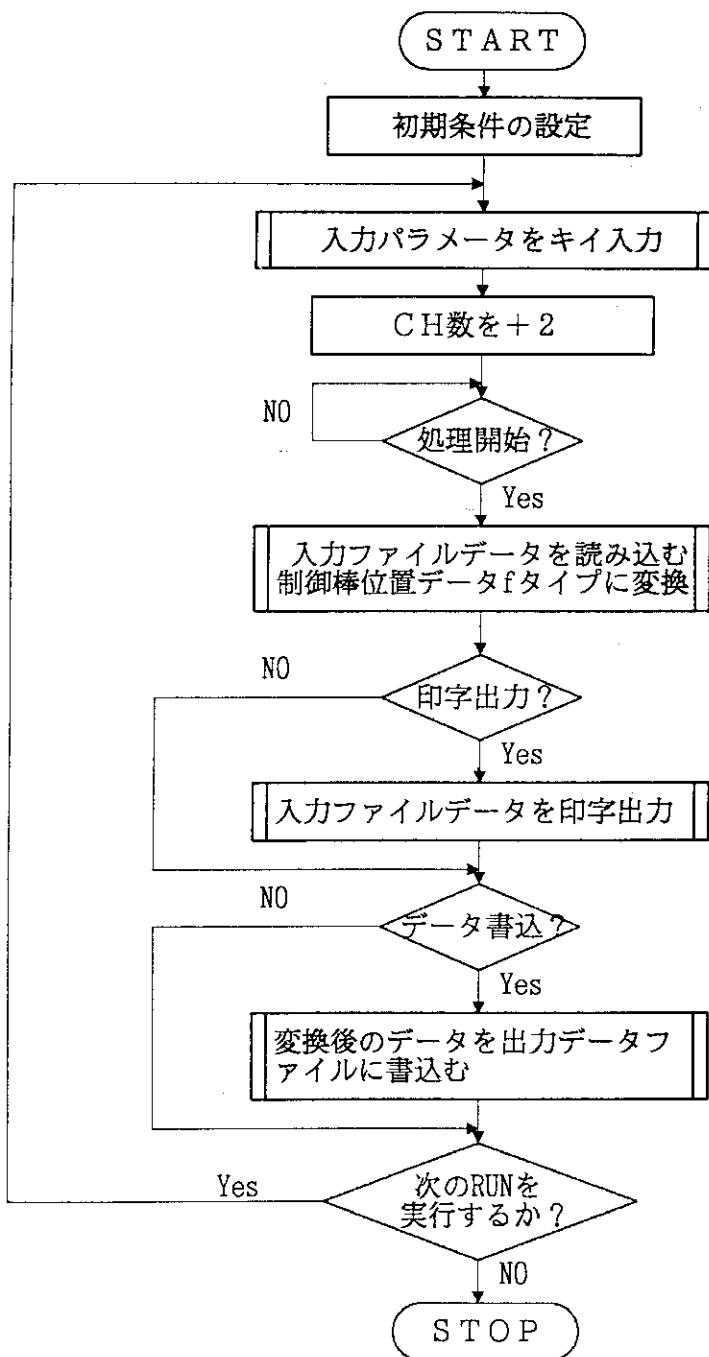
- (1) プログラム名 : VHTRCONV.C
- (2) 入力データファイル名 : キイボードから入力可能
(16文字以内)
- (3) 出力データファイル名 : キイボードから入力可能
(16文字以内)
- (4) データ容量 : 最大30,000データ/CHまで入力可能
- (5) 入力データファイルのチャンネル数 : キイボードから入力可能(6CH以内)
- (6) 出力データファイルのチャンネル数 : 自動的に入力データファイルのチャンネル数+2になる
- (7) 入力データファイルのデータフォーマット : アナログデータ:fタイプ, 柄数は制限なし
バイナリーデータ:dタイプ
- (8) 出力データファイルのデータフォーマット : 10.5fタイプ
- (9) 印字出力選択機能 : コード変換後のデータ
- (10) 処理パラメータの入力方式 : メニュー方式
- (11) 制御棒位置データを使用するか否か : 使用/使用せず
の条件設定 (使用せずを設定した場合は制御棒位置データはすべて0に置き換えられる)
- (12) プログラム再実行の選択機能 : 有り

2.3 プログラムのフローチャート

制御棒位置データ変換プログラムのフローチャートをA2.1図に示す。

2.4 プログラムのコード

制御棒位置データ変換プログラムのコードリストをリスト2.1に示す。



A 2.1 図 制御棒位置データ変換プログラムのフローチャート

リスト2.1 制御棒位置データ変換プログラム

```

*****
*
*      VHTRC Data Code Conversion Program < File name : vhtrconv.c >
*
*          Created      1995.9.6
*          Last Modified 1995.9.8
*          Programed   Y. Fujii
*          Compiled by SPARC2
*
*****
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <sys/file.h>
#include <math.h>
#include <string.h>
#define CHAN_MAX           8          /* Channel Max */
#define CHAN_MIN           1          /* Channel Min */
#define DATA_MAX            30000     /* No of Data Max/Ch */
#define DATA_MIN            1          /* No of Data Min/Ch */
static int    channel;                /* No of Channel */
static int    new_channel;           /* No of New Channel */
static int    max_data;              /* Max No of Data/CH in Original File */
static int    sw1;                  /* Next Run Check Switch */
static int    sw2;                  /* Run Switch */
static int    sw3;                  /* Software Switch */
static int    sw4;                  /* Rod Position Use/Not Use Switch */
static float  *px;                 /* File Name of Original Data */
static int    *pz;                 /* File Name of Output Data */
static struct params {
    char fname1[16];
    char fname2[16];
}
pa={ "vhtrms1.dat",
      "vhtrmscl.dat"
};
static float  xbuff[30000][8];
static int    zbuff[4];

main()
{
    char id[10];
    sw1 = 0;
    sw2 = 0;
    max_data = 6801;
    channel = 6;                      /* Default Max Data/Ch */
    channel = 6;                      /* Default No of Channels */
    while (sw1 == 0){                  /* Check Next Run */
        sw4 = 0;
        params_in();                  /* Default Software Switch 4 = 0 */
        /* Parameter Input */
        sw2 = 1;
        new_channel = channel + 2;   /* New Channel No. */
        px = xbuff[0];               /* Pointer Set */
        pz = zbuff;                  /* Pointer Set */
        while(1)                     /* Start Check */
        {
            printf("start ? (y) : ");
            scanf("%s",id);
            if(id[0] == 'y' || id[0] == 'Y') break;
        }
        read_origin_data();          /* Read Original Data File */
        print_origin_data();          /* Printout Original Data File */
        write_conv_data();            /* Store Converted Data */
        repeat_check();              /* Select Next Run */
    }
}
***** END OF MAIN PROGRAM *****

```

```

***** Input Parameters Subroutine *****
params_in()
{
    char    *pb;
    char    id[10];
    sw3 = 0;
    while (sw3 == 0){
        system("clear");           /* Clear CRT Display */
        printf("Yn");
        printf("***** INPUT PARAMETER *****YnYn");
    /** KEYIN DATA FILE NAME ***/
        pb = "Original Data File Name ? (<16char)      : ";
        string_keyin(pb, pa.fname1);
        pb = "Output Data File Name ? (<16char)      : ";
        sw2 = 0;
        string_keyin(pb, pa.fname2);
    /** SET No. of CHANNELS ***/
        pb = "No of Channels of Original Data File(1 ~ 8) : ";
        one_int_keyin(pb, &channel, CHAN_MAX, CHAN_MIN);
    /** SET MAX DATA/CH ***/
        pb = "No of Data/Ch in Original File (1 ~ 30000) : ";
        one_int_keyin(pb, &max_data, DATA_MAX, DATA_MIN);
    /** SELECT ROD POSITION SIGNAL CONDITION ***/
        printf("Rod Position Signal Data use ? (y/n)      : ");
        scanf("%s", id);
        if(id[0] == 'y' || id[0] == 'Y'){
            sw4 = 0;
        }else{
            sw4 = 1;
        }
        printf("Yn");
        printf("***** INPUT PARAMETERS *****Yn");
        printf("Original Data File Name      : %sYn", pa.fname1);
        printf("Output Data File Name       : %sYn", pa.fname2);
        printf("No of Channels of Original File : %dYn", channel);
        printf("No of Data/Ch in Original File  : %dYn", max_data);
        if(sw4 == 0){
            printf("Rod Position Signal Data      : use Yn");
        }else{
            printf("Rod Position Signal Data      : not use Yn");
        }
        printf("*****YnYn");
        printf("Parameter Change Again ? (y/n) : ");
        scanf("%s", id);
        if(id[0] == 'y' || id[0] == 'Y'){
            sw3 = 0;
            sw2 = 1;
        }else{
            sw3 = 1;
        }
    }
}

***** Get One Integer from Keyboard Subroutine *****
one_int_keyin(b, int_p, max, min)
{
    char    *b;
    int     *int_p;
    int     max, min;
    int     x;
}

```

```

char    s[18];
while(1) {
    printf("%s",b);
    printf("%d ",*int_p);
    fflush(stdin);
    gets(s);
    if(s[0] == 'Y0')break;
    x = atoi(s);
    if(x < min || x > max)
        continue;
    *int_p = x;
    break;
}
}

/******
*          String Keyin Subroutine
*****/
string_keyin(b, char_p)
char    *b,*char_p;
{
    char    s[18];
    int     i;
    while(1) {
        printf("%s",b);
        printf("%s ",char_p);
        fflush(stdin);
        if(sw2 != 0) gets(s);
        gets(s);
        if(s[0] == 'Y0' | s[1] == 'Y0') break;
        for(i=0; s[i] != 'Y0'; i++);
        if(i > 16) continue;
        strcpy(char_p,s);
        break;
    }
}

/******
*          Get Data from Original Data File Subroutine
*****/
read_origin_data()
{
    FILE   *fp,*fopen();           /* File Pointer */
    int    i,j,k,nl;
    int    sum;
    int    multi_fact;
    float  float_sum;
    if((fp = fopen(pa.fname1,"r")) == NULL) {
        printf("open error\n");
        exit();
    }
    for (i=0; i<max_data; i++) {
        for (j=0; j<channel; j++) {
            fscanf(fp,"%f",px);
            px++;
        }
        for (k=0; k<2; k++) {
            sum = 0;
            multi_fact = 1;
            for (nl=1; nl<=4; nl++) {
                pz = zbuff;
                fscanf(fp,"%d",pz);
                if(zbuff[0] == 0){
                    sum = sum + (8 * multi_fact);
                }
                pz++;
            }
        }
    }
}

```

```

        fscanf(fp,"%d",pz);
        if(zbuff[1] == 0){
            sum = sum + (4 * multi_fact);
        }
        pz++;
        fscanf(fp,"%d",pz);
        if(zbuff[2] == 0){
            sum = sum + (2 * multi_fact);
        }
        pz++;
        fscanf(fp,"%d",pz);
        if(zbuff[3] == 0){
            sum = sum + multi_fact;
        }
        multi_fact = multi_fact * 10;
    }
    float_sum = sum;
    if(sw4 == 0){
        *px = float_sum/10.0;
    }else{
        *px = 0.0;
    }
    px++;
}
px = px + (8 - new_channel);
}
fclose(fp);
}

/********************* Original Data Printout Subroutine *****/
print_origin_data()
{
    int e,f;                      /* Keyin Buffer */
    int i,j,k;                    /* Index for Loop */
    printf("Printout Original Data ? (y/n) : ");
    e = getchar();
    f = getchar();
    if(e == 'y' | f == 'y'){
        for(i=0; i<max_data; i++){
            for (j=0; j<channel; j++){
                printf("%10.5f",xbuff[i][j]);
            }
            for (j=0; j<2; j++){
                k = channel + j;
                printf("%10.1f",xbuff[i][k]);
            }
            printf("\n");
        }
    }
}

/********************* Converted Data Store Subroutine *****/
write_conv_data()
{
    FILE *fp,*fopen();           /* File Pointer */
    int e,f;
    int i,j;
    printf("Store Converted Data ? (y/n) : ");
    fflush(stdin);
    e = getchar();
    f = getchar();
    if(e == 'y' | f == 'y'){

```

```
if((fp = fopen(pa.fname2, "w")) == NULL) {
    printf("open error\n");
    exit();
}
for (i=0; i<=max_data; i++) {
    for (j=0; j<new_channel; j++) {
        fprintf(fp, "%10.5f", xbuff[i][j]);
    }
    fprintf(fp, "\n");
}
fclose(fp);
}

/*
 *          Repeat Check Subroutine
 */
repeat_check()
{
    int e, f;                      /* Keyin Buffer */
    printf("\n");
    printf("Repeat Next Run ? (y/n) : ");
    e = getchar();
    f = getchar();
    if (e=='y' || f=='y') {
        sw1 = 0;
    } else {
        sw1 = 1;
    }
}

***** END OF FILE *****
```

付録3 測定チャンネルデータ選択プログラム

3.1 概要

付録2の制御棒位置データ変換プログラムで処理されたデータファイルのフォーマットは、時間軸データを含めたアナログデータチャンネル6CH+制御棒位置信号2CHの構成になっている。この測定チャンネルデータ選択プログラムは、これらのチャンネルデータの中から任意の2つのチャンネルのデータを選択し、新たに1つのデータファイルを作成する機能を備えている。本プログラムもSPARC2ワークステーションを用いてコンパイルした。

3.2 プログラムの仕様

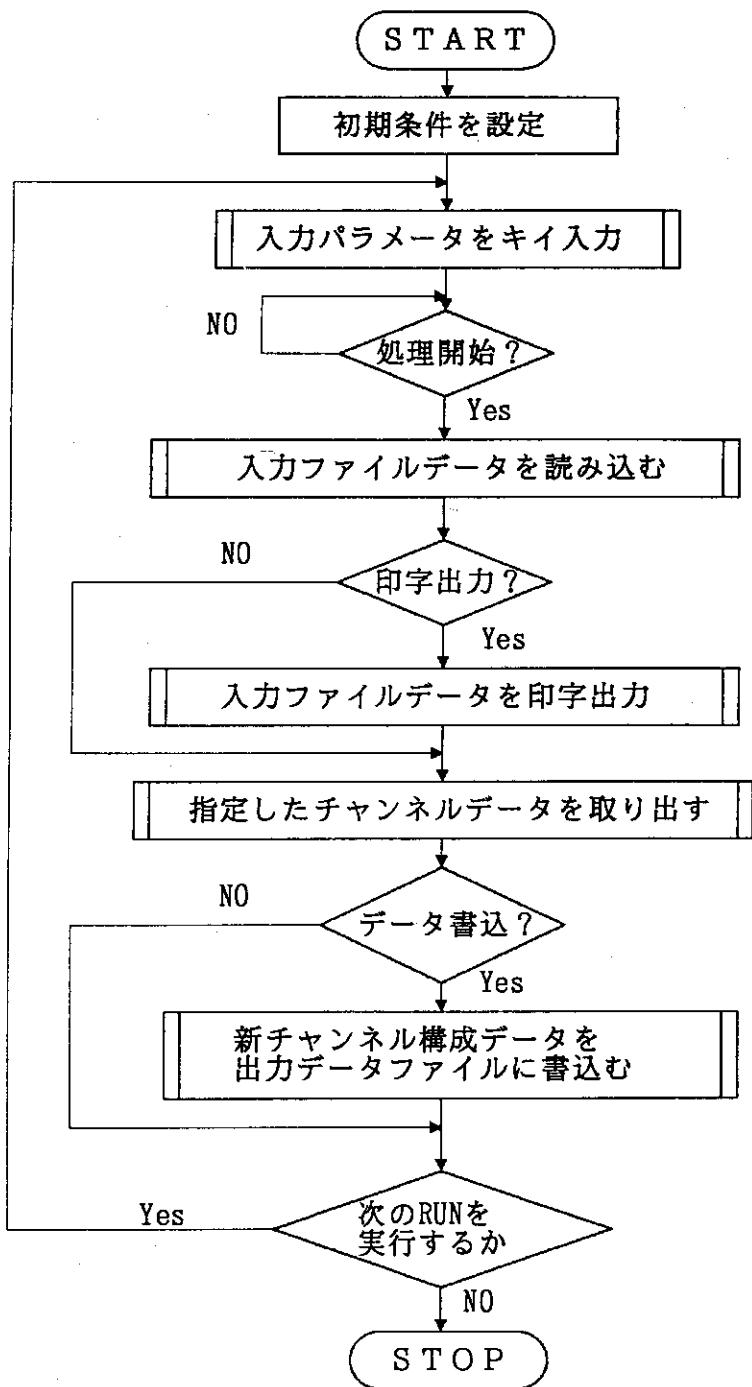
- | | |
|-------------------------|---------------------------|
| (1) プログラム名 | : SELECTCH.C |
| (2) 入力データファイル名 | : キイボードから入力可能
(16文字以内) |
| (3) 出力データファイル名 | : キイボードから入力可能
(16文字以内) |
| (4) データ容量 | : 最大320,000データ/CHまで入力可能 |
| (5) 入力データファイルのチャンネル数 | : キイボードから入力可能(17CH以内) |
| (6) 出力データファイルのチャンネル数 | : 2チャンネル |
| (7) 入力データファイルのデータフォーマット | : アナログデータ :fタイプ, 柄数は制限なし |
| (8) 出力データファイルのデータフォーマット | : 10.5fタイプ |
| (9) 印字出力選択機能 | : 新たなチャンネル構成後のデータ |
| (10) 処理パラメータの入力方式 | : メニュー方式 |
| (11) プログラム再実行の選択機能 | : 有り |

3.3 プログラムのフローチャート

測定チャンネルデータ選択プログラムのフローチャートをA3.1図に示す。

3.4 プログラムのコード

測定チャンネルデータ選択プログラムのコードリストをリスト3.1に示す。



A 3.1 図 チャンネルデータ選択プログラムのフローチャート

リスト 3.1 測定チャンネルデータ選択プログラム

```

*****+
*          Channel Data Select Program < File name : selectch.c >
*          Created      1995.9.25
*          Last Modified 1995.9.27
*          Programed   Y.Fujii
*          Compiled by SPARC2
*
*****+
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <sys/file.h>
#include <math.h>
#include <string.h>
#define CHAN_MAX      17           /* Channel Max */
#define CHAN_MIN      1            /* Channel Min */
#define DATA_MAX      320000        /* No of Data Max/Ch */
#define DATA_MIN      1            /* No of Data Min/Ch */
static int    channel;           /* No of Channel */
static int    select_1st_ch;     /* 1'st Select CH */
static int    select_2nd_ch;     /* 2'nd Select CH */
static int    max_data;         /* Max No of Data/CH */
static int    total_data;       /* Total Data */
static int    sw1;              /* Next Run Check Switch */
static int    sw2;              /* Run Switch */
static int    sw3;              /* Software Switch */
static char   ss[20];          /* Keyin Buffer */
static float  *px;             /* Pointer Set */
static float  *py;             /* Pointer Set */
static struct params {
    char fname1[16];          /* File Name of Original Data */
    char fname2[16];          /* File Name of Output Data */
} pa={ "vhtmsc7.dat",
       "vhtrc07.dat"
     };
static float  xbuff[320000][17];
static float  ybuff[320000][2];

main()
{
    char id[10];
    sw1 = 0;
    sw2 = 0;
    max_data = 10;                /* Default Max Data/Ch */
    channel = 17;                 /* Default No of Channels */
    select_1st_ch = 1;             /* Default 1'st Select CH No */
    select_2nd_ch = 2;             /* Default 2'nd Select CH No */
    while (sw1 == 0) {            /* Check Next Run */
        params_in();              /* Parameter Input */
        sw2 = 1;
        px = xbuff[0];            /* Pointer Set */
        py = ybuff[0];            /* Pointer Set */
        total_data = max_data * channel; /* Start Check */
        while(1)                  /* Start Check */
        {
            printf("start ? [y] : ");
            scanf("%s",id);
            if(id[0] == 'y' || id[0] == 'Y') break;
        }
        read_origin_data();        /* Read Original Data */
        print_origin_data();
        select_data_ch();
        write_select_data();       /* Store Select Data */
        repeat_check();            /* Select Next Run */
    }
}

```

```

        }

}

***** END OF MAIN PROGRAM *****

***** Input Parameters Subroutine *****
***** params_in() *****
{
    char *pb;
    char id[10];
    int e, f;
    sw3 = 0;
    while (sw3 == 0){
        system("clear");           /* Clear CRT Display */
        printf("Yn");
        printf("***** INPUT PARAMETER *****YnYn");
        /*** KEYIN DATA FILE NAME ***/
        pb = "Original Data File Name ? (<16char) : ";
        string_keyin(pb, pa.fname1);
        pb = "Output Data File Name ? (<16char) : ";
        sw2 = 0;
        string_keyin(pb, pa.fname2);
        /*** SET USE CHANNEL ***/
        pb = "No of Channels of Original Data File(1 - 17) : ";
        one_int_keyin(pb, &channel, CHAN_MAX, CHAN_MIN);
        /*** SET 1'st SELECT CH NUMBER ***/
        pb = "1'st Select Channel No (1 - 17) : ";
        one_int_keyin(pb, &select_1st_ch, CHAN_MAX, CHAN_MIN);
        /*** SET 2'nd SELECT CH NUMBER ***/
        pb = "2'nd Select Channel No (1 - 17) : ";
        one_int_keyin(pb, &select_2nd_ch, CHAN_MAX, CHAN_MIN);
        /*** MAX DATA/CH ***/
        pb = "No of Data/Ch (1 - 320000) : ";
        one_int_keyin(pb, &max_data, DATA_MAX, DATA_MIN);
        printf("Yn");
        printf("***** INPUT PARAMETERS *****Yn");
        printf("Original Data File Name      : %sYn", pa.fname1);
        printf("Output Data File Name       : %sYn", pa.fname2);
        printf("Use Channel of Original File : %dYn", channel);
        printf("1'st Select Channel No     : %dYn", select_1st_ch);
        printf("2'nd Select Channel No     : %dYn", select_2nd_ch);
        printf("No of Data/Ch              : %dYn", max_data);
        printf("***** ***** ***** ***** ***** *****YnYn");
        printf("Parameter Change Again ? (y/n) : ");
        scanf("%s", id);
        if(id[0] == 'y' || id[0] == 'Y'){
            sw3 = 0;
            sw2 = 1;
        }else{
            sw3 = 1;
        }
    }
}

***** Get One Integer from Keyboard Subroutine *****
***** one_int_keyin(b, int_p, max, min) *****
{
    char *b;
    int *int_p;
    int max, min;
    {
        int x;

```

```

char    s[16];
while(1){
    printf("%s",b);
    printf("%d ",*int_p);
    fflush(stdin);
    gets(s);
    if(s[0] == 'Y0')break;
    x = atoi(s);
    if(x < min || x > max)
        continue;
    *int_p = x;
    break;
}
}

/***** String Keyin Subroutine ****/
string_keyin(b, char_p)
char    *b, *char_p;
{
    char    s[16];
    int     i;
    while(1){
        printf("%s",b);
        printf("%s ",char_p);
        fflush(stdin);
        if(sw2 != 0) gets(s);
        gets(s);
        if(s[0] == 'Y0' | s[1] == 'Y0') break;
        for(i=0; s[i] != 'Y0'; i++);
        if(i > 16) continue;
        strcpy(char_p,s);
        break;
    }
}

/***** Get Data from Original Data File Subroutine ****/
read_origin_data()
{
    FILE   *fp,*fopen();           /* File Pointer */
    int    i,j;
    if((fp = fopen(pa.fname1,"r")) == NULL) {
        printf("open error\n");
        exit();
    }
    for (i=0; i<max_data; i++){
        for (j=0; j<channel; j++){
            fscanf(fp,"%f",px);
            px++;
        }
        px = px + (17 - channel);
    }
    fclose(fp);
}

/***** Original Data Printout Subroutine ****/
print_origin_data()
{
    int    e,f;                  /* Keyin Buffer */
    int    i,j,k;                /* Index for Loop */
    printf("Printout Original Data ? (y/n) : ");
}

```

```

e = getchar();
f = getchar();
if(e == 'y' | f == 'y'){
    for(i=0; i<max_data; i++){
        for (j=0; j<channel; j++) {
            k = j + 1;
            if(k <= 10 ){
                printf("%10.5f", xbuff[i][j]);
                if(k == 10 )
                    printf("\n");
            }
            else{
                printf("%10.5f", xbuff[i][j]);
                if(k*10 == 0)
                    printf("\n");
            }
        }
        printf("\n");
    }
}

/*********************************************
*                               Select Data Subroutine
*****/



select_data_ch()
{
    int      i,j1,j2;                      /* Index */
    j1 = select_1st_ch - 1;
    j2 = select_2nd_ch - 1;
    for(i=0; i<max_data; i++){
        ybuff[i][0] = xbuff[i][j1];
        ybuff[i][1] = xbuff[i][j2];
    }
}

/*********************************************
*                               Data Store Subroutine
*****/



write_select_data()
{
    FILE   *fp,*open();                  /* File Pointer */
    int     e,f;
    int     i;
    printf("Store Select Ch Data ? (y/n) : ");
    fflush(stdin);
    e = getchar();
    f = getchar();
    if(e == 'y' | f == 'y'){
        if((fp = fopen(pa.fname2,"w")) == NULL) {
            printf("open error\n");
            exit();
        }
        for (i=0; i<max_data; i++){
            fprintf(fp,"%10.5f%10.5f\n", ybuff[i][0], ybuff[i][1]);
        }
        fclose(fp);
    }
}

/*********************************************
*                               Repeat Check Subroutine
*****/



repeat_check()
{
    int     e,f;                         /* Keyin Buffer */

```

```
printf("Yn");
printf("Repeat Next Run ? (y/n) : ");
e = getchar();
f = getchar();
if (e=='y' || f=='y'){
    sw1 = 0;
}else{
    sw1 = 1;
}
*******/

***** END OF FILE *****
```

付録4 制御棒位置・反応度データ変換 プログラム

4.1 概要

付録2の制御棒位置データ変換プログラムで処理されたデータファイルのフォーマットは、時間軸データを含めたアナログデータチャンネル6CH+制御棒位置信号2CHの構成になっている。この制御棒位置・反応度データ変換プログラムは移動側制御棒位置データから反応度変化量を求め、その演算結果をデータファイルの最後の位置に追加する機能を備えている。このプログラムの実行にあたり、入力パラメータを変更することにより、VHTRCのデータロガーで収録した5インチの制御棒位置データについても反応度変化量を求めることができる。また、このプログラムはデータロガーで収録したデータに付されているサンプリング・インディックスNo.を、サンプリング時間軸データに置き換える機能をも備えており、その処理の有無を選択できる。

本プログラムもSPARC2ワークステーションを用いてコンパイルした。

4.2 プログラムの仕様

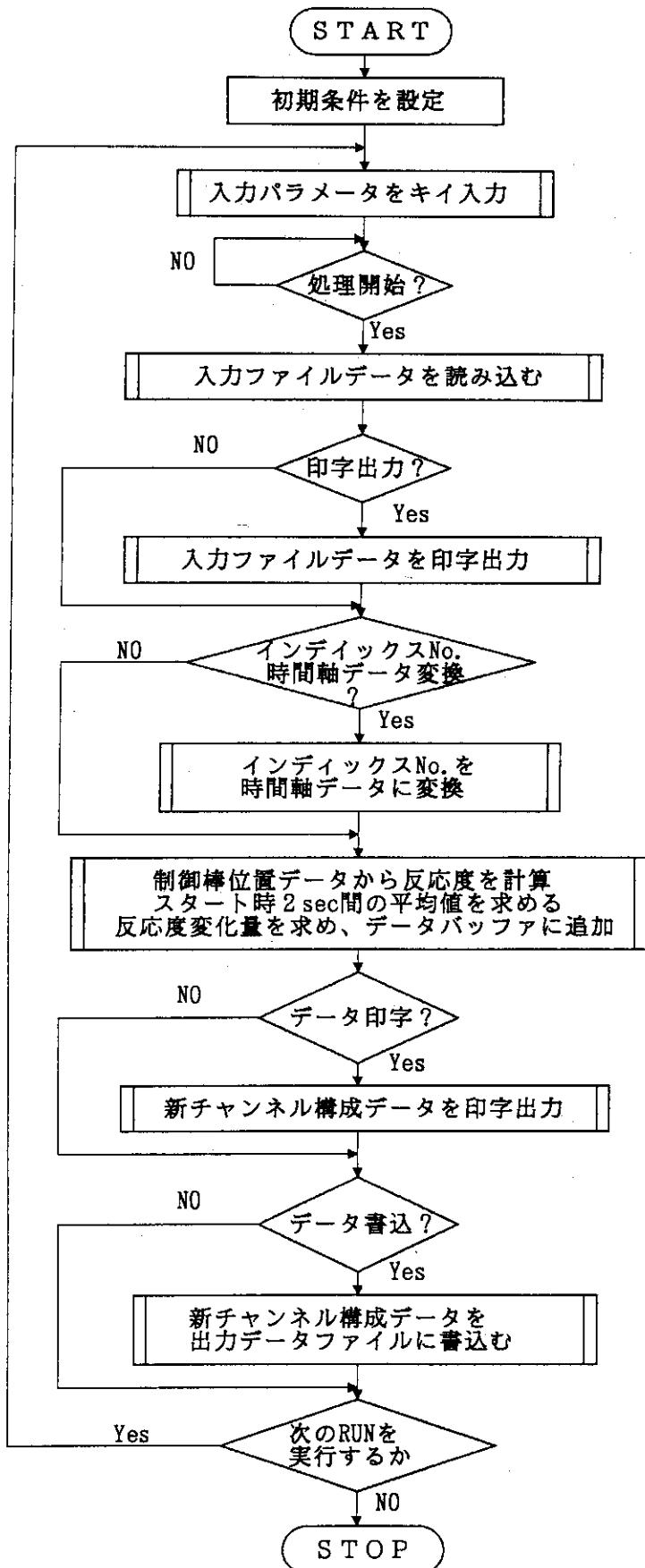
- | | |
|-------------------------|-----------------------------------|
| (1) プログラム名 | : CENTCONV.C |
| (2) 入力データファイル名 | : キイボードから入力可能
(16文字以内) |
| (3) 出力データファイル名 | : キイボードから入力可能
(16文字以内) |
| (4) データ容量 | : 最大50,000データ/CHまで入力可能 |
| (5) 入力データファイルのチャンネル数 | : キイボードから入力可能(8CH以内) |
| (6) 出力データファイルのチャンネル数 | : 入力データファイルのチャンネル数
+1チャンネル |
| (7) 入力データファイルのデータフォーマット | : アナログデータ :fタイプ, 柄数は
制限なし |
| (8) 出力データファイルのデータフォーマット | : 10.5fタイプ |
| (9) 印字出力選択機能 | : オリジナル・データの印字
新たなチャンネル構成後のデータ |
| (10) 制御棒位置データチャンネルの選択 | : 任意CH No. 設定可能 |
| (11) 反応度初期値の平均化 | : 2 sec間の平均値を0とする |
| (12) 反応度の表示単位 | : スタート時点からの反応度変化量を
%単位で表示 |
| (13) 反応度変換対象制御棒 | : 移動側制御棒 |
| (14) 処理パラメータの入力方式 | : メニュー方式 |
| (15) プログラム再実行の選択機能 | : 有り |

4.3 プログラムのフローチャート

制御棒位置・反応度データ変換プログラムのフローチャートをA4.1図に示す。

4.4 プログラムのコード・リスト

制御棒位置・反応度データ変換プログラムのコード・リストをリスト4.1に示す。



A 4.1 図 制御棒位置・反応度変換プログラムのフローチャート

リスト 4.1 制御棒位置・反応度データ変換プログラム

```
*****
*
*      VHTRC Rod Worth Compensate Program < File name : centconv.c >
*
*      Created      1995.10.13
*      Last Modified 1995.10.16
*      Programmed   Y. Fujii
*      Compiled by  SPARC2
*
*****
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <sys/file.h>
#include <math.h>
#include <string.h>
#define CHAN_MAX          8          /* Channel Max */
#define CHAN_MIN          1          /* Channel Min */
#define DATA_MAX          50000     /* No of Data Max/Ch */
#define DATA_MIN          1          /* No of Data Min/Ch */
#define SAMPT_MAX         1000.0    /* Max Sampling Time (sec) */
#define SAMPT_MIN         0.0001   /* Min Sampling Time (sec) */
#define CHANNEL           8          /* No of Channel */
#define NEW_CHANNEL       1          /* No of New Channel */
#define ROD_POSITION_CH  1          /* Rod Position CH */
#define MAX_DATA_CH      1          /* Max No of Data/CH */
#define NEXTRUN_SWITCH   1          /* Next Run Check Switch */
#define RUN_SWITCH        1          /* Run Switch */
#define SW3                1          /* Software Switch */
#define SW4                1          /* Software Switch */
#define KEYIN_BUFFER      20         /* Keyin Buffer */
static int    channel;
static int    new_channel;
static int    rod_position_ch;
static int    max_data;
static int    sw1;
static int    sw2;
static int    sw3;
static int    sw4;
static char   ss[20];
static float  *px;
static float  *py;
static float  sampling_time;
static struct params {
    char   fname1[16];
    char   fname2[16];
}
pa={ "vhtrod1.dat",
      "vhcent01.dat"
    };
static float  xbuff[50001][9];
static float  ybuff[50001];
static float  average_value;

main()
{
    char   id[10];
    sw1 = 0;
    sw2 = 0;
    sampling_time = 0.2;           /* Default Sampling Time (sec) */
    max_data = 10;                /* Default Max Data/Ch */
    channel = 3;                  /* Default No of Channels */
    rod_position_ch = 2;          /* Default Rod Position CH No */
    while (sw1 == 0){             /* Check Next Run */
        params_in();              /* Parameter Input */
        sw2 = 1;                  /* New Channel No. */
        new_channel = channel + 1; /* Pointer Set */
        px = xbuff[0];            /* Pointer Set */
        py = ybuff;                /* Start Check */
        while(1)
        {
            printf("start ? [y] : ");
            scanf("%s",id);
            if(id[0] == 'y' || id[0] == 'Y') break;
        }
        read_origin_data();        /* Read Original Data */
    }
}
```

```

print_origin_data();
if(sw4 == 1){
    time_data_modify();
}
data_compensate();           /* Rod Position Convert to CENT */
print_compensate_data();     /* Printout Compensate Data */
write_compensate_data();     /* Store Compensate Data */
repeat_check();              /* Select Next Run */
}

*******/

***** END OF MAIN PROGRAM *****

/*****
*          Input Parameters Subroutine
*****
params_in()
{
    char    *pb;
    char    id[10];
    int     e,f;
    sw3 = 0;
    while (sw3 == 0){
        system("clear");           /* Clear CRT Display */
        printf("Yn");
        printf("***** INPUT PARAMETER *****YnYn");
        /*** KEYIN DATA FILE NAME ***/
        pb = "Original Data File Name ? (<16char)      : ";
        string_keyin(pb, pa.fname1);
        pb = "Output Data File Name ? (<16char)      : ";
        sw2 = 0;
        string_keyin(pb, pa.fname2);
        /*** SET USE CHANNEL ***/
        pb = "No of Channels of Original Data File(1 ~ 8) : ";
        one_int_keyin(pb, &channel, CHAN_MAX, CHAN_MIN);
        /*** SET SELECT CH NUMBER ***/
        pb = "Select Channel No of Rod Position(1 ~ 8)   : ";
        one_int_keyin(pb, &rod_position_ch, CHAN_MAX, CHAN_MIN);
        /*** SET DATA SAMPLING TIME ***/
        pb = "Data Sampling Time (0.0001~1000.0 sec)   : ";
        float_keyin(pb, &sampling_time, SAMPT_MAX, SAMPT_MIN);
        /*** MAX DATA/CH ***/
        pb = "No of Data/Ch (1 ~ 50000)                 : ";
        one_int_keyin(pb, &max_data, DATA_MAX, DATA_MIN);
        /*** SET TIME DATA MODIFY CONDITION ***/
        printf("Time Data Modify ? (y/n)                  : ");
        scanf("%s", id);
        if(id[0] == 'y' || id[0] == 'Y'){
            sw4 = 1;
        }else{
            sw4 = 0;
        }
        printf("Yn");
        printf("***** INPUT PARAMETERS *****Yn");
        printf("Original Data File Name      : %sYn",pa.fname1);
        printf("Output Data File Name       : %sYn",pa.fname2);
        printf("Use Channel of Original File: %dYn",channel);
        printf("Channel No of Rod Position  : %dYn",rod_position_ch);
        printf("No of Data/Ch                : %dYn",max_data);
        printf("Data Sampling Time (sec)    : %3.5fYn",sampling_time);
        if(sw4 == 1){
            printf("Time Data Modify           : YES Yn");
        }else{
            printf("Time Data Modify           : NO Yn");
        }
}

```

```

printf("*****\n");
printf("Parameter Change Again ? (y/n) : ");
scanf("%s", id);
if(id[0] == 'y' || id[0] == 'Y'){
    sw3 = 0;
    sw2 = 1;
} else{
    sw3 = 1;
}
}

/*****
*          Get One Integer from Keyboard Subroutine
*****
one_int_keyin(b, int_p, max, min)
char    *b;
int     *int_p;
int     max, min;
{
    int    x;
    char   s[16];
    while(1){
        printf("%s", b);
        printf("%d ", *int_p);
        fflush(stdin);
        gets(s);
        if(s[0] == 'Y0')break;
        x = atoi(s);
        if(x < min || x > max)
            continue;
        *int_p = x;
        break;
    }
}

/*****
*          String Keyin Subroutine
*****
string_keyin(b, char_p)
char    *b, *char_p;
{
    char   s[16];
    int    i;
    while(1){
        printf("%s", b);
        printf("%s ", char_p);
        fflush(stdin);
        if(sw2 != 0) gets(s);
        gets(s);
        if(s[0] == 'Y0' || s[1] == 'Y0') break;
        for(i=0; s[i] != 'Y0'; i++);
        if(i > 16) continue;
        strcpy(char_p, s);
        break;
    }
}

/*****
*          Get One Float from Keyboard Subroutine
*****
float_keyin(b, float_p, max, min)
char    *b;
float   *float_p, max, min;
{
    double   x, atof();

```

```

char          s[16];
while(1){
    printf("%s", b);
    printf("%10.5f ", *float_p);
    gets(s);
    if(s[0]=='Y') break;
    x=atof(s);
    if(x<min || x>max)
        continue;
    *float_p=(float)x;
    break;
}
}

***** Get Data from Original Data File Subroutine ****
read_origin_data()
{
    FILE    *fp,*fopen();           /* File Pointer */
    int     i,j;
    if((fp = fopen(pa.fname1,"r")) == NULL) {
        printf("open error\n");
        exit();
    }
    for (i=0; i<max_data; i++) {
        for (j=0; j<channel; j++) {
            fscanf(fp,"%f",px);
            px++;
        }
        px = px + (g - channel);
    }
    fclose(fp);
}

***** Original Data Printout Subroutine ****
print_origin_data()
{
    int    e,f;                  /* Keyin Buffer */
    int    i,j;                  /* Index for Loop */
    printf("Printout Original Data ? (y/n) : ");
    e = getchar();
    f = getchar();
    if(e == 'y' | f == 'y'){
        for(i=0; i<max_data; i++){
            for (j=0; j<channel; j++){
                printf("%10.5f",xbuff[i][j]);
            }
            printf("\n");
        }
    }
}

***** Time Data Modify Subroutine ****
time_data_modify()
{
    int    i;                    /* Index for Loop */
    float fi;
    for(i=0; i<max_data; i++){
        fi = i;
        xbuff[i][0] = sampling_time * fi;
    }
}

```

```

}

/*
*          Compensate Data Subroutine
*/
data_compensate()
{
    int     i,j;                      /* Index */
    double  x1,x2,x3,x4;
    double  a1,a2,a3,a4,a5;
    float   reactivity;
    a1 = 0.11822e+3;
    a2 = -0.24733;
    a3 = 0.80427e-4;
    a4 = 0.11144e-6;
    a5 = -0.62794e-10;
    j = rod_position_ch - 1;
    for(i=0; i<max_data; i++){
        x1= xbuff [i][j];
        x2 = x1 * x1;
        x3 = x2 * x1;
        x4 = x3 * x1;
        reactivity = -(a2 * x1)-(a3 * x2)-(a4 * x3)-(a5 * x4);
        ybuff[i] = reactivity;
    }
    data_average();
    for(i=0; i<max_data; i++){
        xbuff[i][new_channel-1] = ybuff[i] - average_value;
    }
}

/*
*          Data Average Subroutine
*/
data_average()
{
    int     i,j;                      /* Index */
    int     times_n;
    float   sum;
    times_n = (2.0 / sampling_time) + 0.2; /* Average Times : 2 sec */
    sum = 0.0;
    for(i=0; i<times_n; i++){
        sum = sum + ybuff[i];
    }
    average_value = sum / times_n;
    printf("Times n = %2d  Average Value = %6.2f\n", times_n, average_value);
}

/*
*          Compensate Data Printout Subroutine
*/
print_compensate_data()
{
    int     e,f;                      /* Keyin Buffer */
    int     i,j;                      /* Index for Loop */
    printf("Printout Compensate Data ? (y/n) : ");
    fflush(stdin);
    e = getchar();
    f = getchar();
    if(e == 'y' | f == 'y'){
        for(i=0; i<max_data; i++){
            for(j=0; j<new_channel; j++){
                printf("%10.5f", xbuff[i][j]);
            }
            printf("\n");
        }
    }
}

```

```

        printf("\n");
    }

/***** Compensate Data Store Subroutine *****
*                                         *
***** *****/
write_compensate_data()
{
    FILE *fp,*fopen();           /* File Pointer */
    int e,f;
    int i,j;
    printf("Store Compensate Data ? (y/n) : ");
    fflush(stdin);
    e = getchar();
    f = getchar();
    if(e == 'y' || f == 'y'){
        if((fp = fopen(pa.fname2, "w")) == NULL) {
            printf("open error\n");
            exit();
        }
        for (i=0; i<max_data; i++){
            for(j=0; j<new_channel; j++){
                fprintf(fp, "%10.5f", xbuff[i][j]);
            }
            fprintf(fp, "\n");
        }
        fclose(fp);
    }
}

/***** Repeat Check Subroutine *****
*                                         *
***** *****/
repeat_check()
{
    int e,f;                      /* Keyin Buffer */
    printf("\n");
    printf("Repeat Next Run ? (y/n) : ");
    e = getchar();
    f = getchar();
    if (e=='y' || f=='y'){
        sw1 = 0;
    }else{
        sw1 = 1;
    }
}

***** END OF FILE *****/

```

付録 5 M A T L A B 作図コマンド プログラムの一例

5.1 概要

本報告書に掲載したデータのグラフはすべて数値計算ライブラリの機能を備えた市販のプログラムMATLAB for Windowsを利用して作成した。MATLABプログラムの備えている機能は膨大なものであるが、今回はそのうちの作図機能のみを使用した。使用した計算機はPC9821ap型パソコンである。MATLABの仕様やその使用方法については、マニュアルを参照されたい。ここでは、今回の実験データの作図用に作成したコマンド・ファイルの一例を参考のために示した。

5.2 コマンド・ファイルの一例

- (1) コマンド・ファイル名 : VTEST1.m
- (2) 入力データファイル名 : VHTMSC1.DAT
(コマンド・ファイル内ではデータ・ファイルの格納されているパス名を付けて指定する)
- (3) データ容量 : [8,13000]
(マトリックス形式でチャンネル数、データ数/CHを設定する)
- (4) 表示色 : 各変数に表示色を指定可能
(ただしグラフ・プリントはすべて黒色になる)
- (5) X軸の値 : 時間軸データによる時間(SEC単位)
- (6) Y軸の値 : 各変数の電圧値(V単位)
- (7) X軸・Y軸のスケーリング : オート・スケーリング
(マニュアル設定も可能)
- (8) 表示サイズ : Windowsの機能で任意拡大縮小可能
ただし、グラフ・プリントのサイズは一定で変更不可能

5.3 コマンド・ファイルのリスト

実験データの作図用に作成したコマンド・ファイル・リストの一例をリスト 5.1 に示す。

リスト5.1 MATLAB作図コマンドプログラムの一例

```
function test7
% test M-file (Read File)
fid=fopen('b:¥matlab¥vhtrc¥vhtmsc7.dat','r')
A=fscanf(fid,'%g',[8,13000]);
status=fclose(fid)
B=A';
T=B(1:12801);
A1=B(1:12801,2:2);
A2=B(1:12801,3:3);
A3=B(1:12801,4:4);
A4=B(1:12801,5:5);
A5=B(1:12801,6:6);
A6=B(1:12801,7:7)/200;
A7=B(1:12801,8:8)/200;
plot(T,A1,'w-')
hold on
plot(T,A2,'r-')
plot(T,A3,'g-')
plot(T,A4,'y-')
plot(T,A5,'c-')
plot(T,A6,'b-')
plot(T,A7,'m-')
title('VHTRC TEST RUN No.7')
xlabel('Time(sec)')
ylabel('Mesurment Value (V)')
hold off
```