LBL--31822

DE92 017085

Isotopic Tracer Studies of Fischer-Tropsch Synthesis over Ru/TiO_2 Catalysts

Kamala Raghunathan Krishna
Ph. D. Thesis

Department of Chemical Engineering
University of California

and

Chemical Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

January 1992

This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098.

Isotopic Tracer Studies of Fischer-Tropsch Synthesis over Ru/TiO_2 Catalysts

by

Kamala Raghunathan Krishna

ABSTRACT

Fischer-Tropsch synthesis is a process in which CO and H_2 react to give predominantly liquid hydrocarbons. The reaction can be considered a special type of polymerization in which the monomer is produced *in situ*, and chain growth occurs by a sequence of independently repeated additions of the monomer to the growing chain. An investigation has been conducted to study the CO hydrogenation reaction in order to better understand catalyst deactivation and the elementary surface processes involved in chain growth.

Isotopic tracers are used in conjunction with transient-response techniques in this study of Fischer-Tropsch synthesis over Ru/TiO_2 catalysts. Experiments are conducted at a total pressure of 1 atmosphere, reaction temperatures of 453-498 K and D_2/CO (or H_2/CO) ratios of 2-5. Synthesis products are analyzed by gas chromatography or isotope-ratio gas chromatography-mass spectrometry.

Ru/TiO_2 catalysts deactivate with no change of product selectivity and the rate of deactivation correlates with initial

catalyst activity. Deactivation occurs at an initial rapid rate, followed by a slower activity loss. Deactivation is accompanied by a loss in CO uptake and the accumulation of various types of carbonaceous species. The long-term loss of activity is attributed to the buildup of long chain hydrocarbon product species.

Rate constants for chain initiation, propagation and termination are evaluated under steady-state reaction conditions by using transients in isotopic composition. The activation energy for chain termination is much higher than that for propagation, accounting for the observed decrease in the chain growth parameter with increasing temperature. Coverages by reaction intermediates are also estimated. The dominant reactive surface species are monomeric building units, which occupy 0.2-0.6 ML. Alkyl species that are the direct hydrocarbon product precursors occupy < 0.2 ML. Adsorbed CO covers 0.7 ML.

When small amounts of ^{12}C-labelled ethylene are added to $^{13}CO/H_2$ synthesis gas, ethylene acts as the sole chain initiator. Ethylene-derived carbon also accounts for 45 % of the C_1 monomer pool.

To

my parents,

my parents-in-law,

Ashok,

and

Ananth.

Table of Contents

List of Figures

Chapter II

Chapter IV

Chapter V

List Of Tables

Chapter II

Chapter IV

Chapter V

List of Symbols

ASF	Anderson-Schulz-Flory
C_α	Carbidic carbon (ML)
C_β	Alkyl carbon chains (ML)
C_β'	Alkyl carbon chains that are precursors to the C_{2+} hydrocarbon products (ML)
C_β''	Unreactive alkyl chains, of longer chain length than C_β' (ML)
F_1	Fraction of labelled carbon in the surface precursor to methane
F_m	Fraction of labelled carbon in the monomer pool
F_n	Fraction of labelled carbon in the pool of carbon chain length n
K	Equilibrium constant for CO adsorption
k	Rate parameter
k_i	Initiation rate constant (s^{-1})
k_p	Intrinsic Rate constant for propagation (s^{-1})
k_p'	Apparent rate constant for propagation (s^{-1})
k_t	Rate constant for termination (s^{-1})
k_1	Rate constant
k_2	Rate constant
k_3	Rate constant
ML	Monolayer, unit of coverage based on hydrogen chemisorption to estimate the number of surface exposed

	Ru atoms
N_{CO}	Turnover frequency of CO (s^{-1}) i.e., moles of CO converted per moles of surface-exposed metal atoms per sec
N_{CO}^{o}	Apparent turnover frequency of CO at t_r = 0 min
N_{C_1}	Turnover frequency of methane (s^{-1})
N_{C_n}	Turnover frequency of products of carbon chain length n (s^{-1})
n	Carbon number
n_{av}	Average carbon number of hydrocarbon products
P_{CO}	Partial pressure of CO
t_r	Time onstream after reaction startup
α	ASF chain growth probability or chain growth parameter
θ	Surface coverage
θ_{CO}	CO coverage (ML)
θ_m	Coverage by monomer building block
θ_1	Coverage by methane precursor
θ_n	Coverage by precursor to hydrocarbon products of chain length n
θ_α	Coverage by $C_\alpha + C_\beta'$ (ML)
$\theta_{\beta''}$	Coverage by C_β'' (ML)
τ	Lifetime of surface alkyl chains
τ_m	Lifetime of monomer pool

Acknowledgements

I would like to thank Dr. Bell for his guidance, interest and constant encouragement throughout the course of my research. I would also like to thank Dr. Theodorou and Dr. Stacy for serving on my dissertation committee.

I am grateful to the members of the Bell group, past and present, for making my stay at Berkeley enjoyable and also for helping me through many crises! A special thanks to Phil, Reha, Rebecca, Gina, Judith, Grant, Rich, Bill, Rick, Mike, Catherine, Narcis, Debbie, Alon, Mark, Larry, Steve, Greg, Kevin, Janet, Ken, Marjorie, Karen, Dean, Bob, Rusty, Nick, Isao, Hiroshi, Takashi, Randy, Craig and Ed.

I would like to thank Wai Ming (Henry) Chan for writing and modifying my GC-MS software and for helping me troubleshoot my electronic equipment woes. Nancy Monroe provided drafting expertise that I am grateful for. Kay and Ferne have helped me with advice and encouragement.

Finally, I am deeply indebted to my parents for their support and encouragement throughout my academic career. My parents-in law have also been a source of strength. Ashok has been a pillar of strength and coped magnificently on the home front. I owe an extra special thank you to Ashok and my son Ananth, who actively kept my spirits up at every stage, and endured my long work hours without (too much) protest.

Chapter I

Introduction

Fischer-Tropsch synthesis is a route for producing liquid fuels from synthesis gas, a mixture of CO and H_2 (1-3). The most attractive potential sources of synthesis gas are the steam reforming of natural gas (methane) and the gasification of coal. Natural gas is often found in remote locations, and conversion to liquid fuel would facilitate storage, transportation and utilization of this resource. Further, since coal is much more abundant than the natural reserves of liquid hydrocarbons, considerable interest in hydrocarbon synthesis from CO and H_2 has been generated by the need for long-range alternative sources of liquid fuels.

The composition of the products of Fischer-Tropsch synthesis depends on the catalyst employed as well as the reaction conditions, in particular, the pressure, the temperature, and the H_2/CO ratio. The predominant products of Fischer-Tropsch synthesis are linear alkanes and alkenes as well as H_2O or CO_2. Aldehydes, alcohols, and carboxylic acids are also formed, but in lower concentrations than hydrocarbons.

The group VIII metals Fe, Co and Ru are active for Fischer-Tropsch catalysis. Promoters and supports can also influence the activity and selectivity of a catalyst. Since a broad product distribution of hydrocarbons, ranging from methane to heavy waxes is produced, it is desirable to enhance the selectivity for liquid

hydrocarbons. In order to accomplish this goal, research efforts are directed towards a better understanding of the mechanism of the reaction and the elementary surface processes involved in the reaction.

The reaction mechanism can be viewed as a surface polymerization process in which the monomer is produced *in situ* from CO and H_2 on the catalyst surface (2). The products of the reaction often follow an Anderson-Schulz-Flory (ASF) distribution, for which $[C_{n+1}]/[C_n] = \alpha$, where α is independent of chain length, n, and $\alpha < 1$. If it is assumed that polymerization occurs via the addition of a C_1 monomer unit to growing hydrocarbon chains, then α represents the probability of chain growth and is given by $\alpha = r_p/(r_p+r_t)$, where r_p and r_t are the rates of chain propagation and chain termination, respectively. Although the effects of process variables such as reaction temperature and H_2/CO ratio on the product distribution (i. e., α) and overall activity have been well-studied, the elementary reaction steps that occur during the reaction are not well-understood.

The objective of the present investigation is to better understand the elementary steps that constitute the CO hydrogenation reaction. One goal is to characterize catalyst deactivation and to try and understand the causes of catalyst decay. Isotopic tracers and temperature-programmed surface reaction (4) are used for this purpose. A second goal is to elucidate the mechanism of chain growth during Fischer-Tropsch synthesis. In particular, it is desirable to obtain the rate constants for chain

initiation, propagation and termination and to determine surface coverages by reaction intermediates. Isotopic tracers are combined with transient-response experiments in order to accomplish this aim, since steady-state rate data can only yield product formation rates and distributions (5, 6). Finally, the effect of ethylene addition to the CO/H_2 Fischer-Tropsch synthesis feed on chain initiation and propagation is investigated. Isotopic labelling is used to differentiate between carbon from CO and carbon from ethylene in the product. This further allows the dynamics of ethylene incorporation to be studied.

Ruthenium supported on titania is used in this work for several reasons. Ruthenium has a high specific activity for Fischer-Tropsch synthesis and has high chain growth probabilities at mild reaction conditions (7). A ruthenium catalyst produces mainly straight-chain hydrocarbons, which are believed to be the primary reaction products. Ruthenium does not produce large amounts of oxygenates, thereby simplifying product identification and analysis. Titania is used to support and disperse the ruthenium since titania-supported Group VIII metals are known to be more active than silica- or alumina-supported Group VIII metals (8-11).

This thesis is divided into four parts. Chapter 2 is a study of catalyst deactivation during CO hydrogenation. The dynamics of catalyst deactivation are determined and the role of the build up of carbonaceous species is assessed. The effects of Ru dispersion, TiO_2 phase and TiO_2 surface area on catalyst activity and selectivity are determined. In Chapter 3, expressions are derived for evaluating

the rate coefficients for chain propagation, k_p, and for chain termination, k_t. Chapter 4 describes the use of the isotopic transient technique to evaluate the rate coefficients for chain initiation, propagation, and termination, and the coverages by reaction intermediates. The effects of reaction temperature and the D_2/CO ratio in the feed on rate coefficients and surface coverages are investigated. Chapter 5 describes the effect on chain growth of adding small amounts of ethylene to the CO/H_2 feed.

REFERENCES

1. Anderson, R. B., 'The Fischer-Tropsch Synthesis', Academic Press, New York, 1984.
2. Biloen, P. and Sachtler, W. M. H., *Adv. Catal.* **30**, 165 (1981).
3. Bell, A. T., *Catal. Rev.-Sci. Eng.* **23**, 203 (1981).
4. Falconer, J. L. and Schwarz, J. A., *Catal. Rev.-Sci. Eng.* **25**, 141 (1983).
5. Bennett, C. O. , p1, *in* "Catalysis under Transient Conditions", ACS Symposium Series, eds. Bell, A. T. and Hegedus, L. L., 1982.
6. Biloen, P., *J. Molec. Catal.* **21**, 17 (1983).
7. Vannice, M. A., *J. Catal.* **37**, 449 (1975).
8. Vannice, M. A. and Garten, R. L., *J. Catal.* **56**, 236 (1979).
9. Vannice, M. A., and Sudhakar, C., *J. Phys. Chem.* **88**, 2429 (1984).
10. Reuel, R. C., and Bartholomew, C. H., *J. Catal.* **85**, 78 (1984).
11. Rieck, J. S., and Bell, A. T., *J. Catal.* **99**, 262 (1986).
12. Jordan, D. S., and Bell, A. T., *J. Phys. Chem.* **90**, 4797 (1986).
13. Kellner, C. S., and Bell, A. T., *J. Catal.* **70**, 418 (1971).

Chapter II

An Isotopic Tracer Study of the Deactivation of Ru/TiO_2 Catalysts During Fischer-Tropsch Synthesis

ABSTRACT

An investigation of the causes of catalyst deactivation during Fischer-Tropsch synthesis over Ru/TiO_2 catalysts has been carried out. The effects of Ru dispersion, TiO_2 phase, and TiO_2 surface area on catalyst activity and selectivity were examined. CO chemisorption capacity and carbon species accumulation were determined as a function of reaction time using isotopic tracer techniques in conjunction with temperature-programmed surface reaction (TPSR). The results of this investigation show that all of the catalysts undergo deactivation with time, with no change in product selectivity. Initially, deactivation is very rapid followed by a slower activity loss. The long-term rate of deactivation is proportional to the initial CO turnover frequency, obtained by extrapolation of the long-term activity data. Deactivation is accompanied by a progressive loss in CO chemisorption capacity as well as an accumulation of various types of carbon species. Carbidic carbon, C_α, and alkyl carbon chains, C_β, were observed to accumulate as reaction proceeds. C_β consists of two species, C_β' which is the precursor to C_{2+} hydrocarbon products and C_β'' which

consists of longer alkyl chains and does not participate in the production of gas phase products. The rapid initial loss in activity in the first ten minutes correlates with the $C_\alpha+C_\beta$ accumulation; the long-term loss in CO uptake and catalyst activity are probably due to C_β". The alkyl chains comprising C_β" do not undergo hydrogenolysis under reaction conditions, probably due to inaccessibility to hydrogen.

INTRODUCTION

A number of previous investigations have shown that while Ru exhibits a high specific activity for Fischer-Tropsch synthesis, it undergoes a progressive loss in activity that is accompanied by a build-up of carbonaceous species (1-6). Dalla Betta and co-workers (1,2) have ascribed the loss in activity of Ru/Al_2O_3 catalysts to the carbonaceous deposits, and have noted that the original activity of these catalysts could be restored by reduction in H_2. Similar observations have been reported by Everson *et al.* (3) and Ekerdt and Bell (4). These authors noted that the rate of deactivation was a function of reaction conditions. Using infrared spectroscopy, Ekerdt and Bell (4) and Yamasaki and co-workers [7] observed the accumulation of hydrocarbon species on the surface under reaction conditions. Ekerdt and Bell (4) concluded that the observed species did not function as reaction intermediates, since these species could only be removed by H_2 reduction in the absence of CO. Reduction of the catalyst following reaction demonstrated that during reaction, the catalyst had accumulated 1-6 Ru ML equivalents of carbon. Based on studies on a Ru/Al_2O_3 catalyst, Bartholomew *et al.* (6) have proposed a model for the rate of deactivation. According to this model, the rate of deactivation is proportional to the partial pressure of CO and the concentration of active sites. Temperature programmed reaction of the catalyst after reaction produced a spectrum exhibiting two peaks. The first of these peaks occurred at 460 K and corresponded to the removal of 5 ML equivalents of

carbon, whereas the second peak occurred at 685 K and corresponded to the removal of 2.5 ML equivalents of carbon.

Isotopic tracer studies conducted in this laboratory (8-14) have demonstrated that two types of carbon are deposited on the surface of Ru particles supported on SiO_2 and TiO_2. The first is carbidic, or C_α, carbon. This species is produced by the dissociation of adsorbed CO and serves as a precursor to methane and C_{2+} hydrocarbons. The second form of carbon consists of short (3-5 carbon atoms) alkyl chains and is designated as C_β. By combining isotopic tracer techniques with temperature programmed reaction, it has been shown that the C_β pool of carbon can be divided into two pools. The first of these pools is designated as C_β', and corresponds to alkyl groups serving as intermediates to the C_{2+} hydrocarbons. The second pool of carbon, designated as C_β'', corresponds to longer alkyl chains which are not involved in the formation of reaction products. Under reaction conditions, the coverages of C_α and C_β' come to a steady state with time, whereas the accumulation of C_β'' increases monotonically. Zhou and Gulari (15) have also found evidence for the presence of C_α and C_β during CO hydrogenation over Al_2O_3-supported Ru. The inventory of C_α was observed to go through a maximum with reaction time, whereas the inventory of C_β increased monotonically. In a subsequent study (16), the authors speculated that the low activity alkyl species that could be observed by infrared spectroscopy were located on sites surrounded

by adsorbed CO.

The objective of this study was to characterize the deactivation of Ru/TiO_2 catalysts during Fischer-Tropsch synthesis and to identify the causes for the loss in activity. The influence of Ru dispersion, TiO_2 phase, and TiO_2 surface area were considered. The CO chemisorption capacity and the quantity of carbon accumulated were determined as a function of time under reaction conditions. Isotopic tracer techniques were used in conjunction with temperature-programmed surface reaction (TPSR) to study changes in the reactivity of CO and the different forms of carbon accumulated on the catalyst.

EXPERIMENTAL

Four titania supports were used for this study: Degussa P-25 titania, a mixture of 30% rutile and 70% anatase; two anatase supports differing in BET surface area; and rutile. These supports will be referred to as $TiO_2(D)$, $TiO_2(A1)$, $TiO_2(A2)$, and $TiO_2(R)$. $TiO_2(A1)$ is the higher surface area anatase.

$TiO_2(A1)$ was prepared as follows. Titanium isopropoxide ($Ti(C_3H_7O)_4$, Dupont) was added dropwise to an excess of isopropyl alcohol at 274 K over a period of 4.5 hours with constant mixing. The resulting slurry was allowed to settle, decanted and washed with distilled water. The decanting and washing was repeated 5-6 times, and the slurry was then filtered and washed several times. The final product was ground and calcined in an O_2 atmosphere. The

preparation of TiO_2(A2) was similar to that of TiO_2(A1), except that the initial step was carried out at 280 K.

TiO_2(R) was prepared according to the procedure described by Kikuchi *et al.* (16). Titanium tetrachloride ($TiCl_4$) was added dropwise to hot distilled water at 368K, stirred for 1 hour, and then cooled to ambient temperature. After filtration, precipitated titanium hydroxide was washed with distilled water and filtered again. The washing cycle was repeated several times and finally the precipitate was dried overnight at 383 K. To remove residual chloride impurities, the support was Soxhlet extracted with water and then calcined in pure O_2 at 673 K. Extraction and calcination were repeated three times (18).

The phase structure of each TiO_2 support was confirmed using Raman spectroscopy, since this technique is very sensitive to the phase of TiO_2 and can detect small proportions of anatase in the presence of rutile. By this means, it was established that TiO_2(R) contained no traces of anatase and that TiO_2(A1) and TiO_2(A2) contained no detectable traces of rutile.

Ru was introduced onto each support by incipient wetness impregnation using an aqueous solution of $RuCl_3.3H_2O$ (Strem). The impregnated supports were dried in air at 373 K overnight, and then sieved to -30, +60 mesh. Reduction of the dried products were carried out at 503 K in a flow of pure H_2. The reduction temperature was intentionally kept low to keep the catalyst in its low-temperature reduced state as much as possible (19). To check for complete conversion of the $RuCl_3$ to metallic Ru, the chloride

content of each catalyst was determined after reduction. The chloride level in $Ru/TiO_2(D)$ and $Ru/TiO_2(A1)$ was below the detection limit (0.02%). For $Ru/TiO_2(A2)$ and $Ru/TiO_2(R)$, the levels were 0.04% and 0.07% respectively. The weight loading of Ru was determined by X-ray fluorescence.

Table 1 summarizes the physical characteristics of each catalyst. The dispersion of the Ru was determined by H_2 chemisorption at 373 K following a two hour reduction at 473 K. Chemisorption was carried out at 373 K because hydrogen adsorption is known to approach equilibrium slowly at room temperature (20), and equilibrium is attained more rapidly at this temperature (21, 22). A static chemisorption apparatus was used. The isotherms for total and reversible adsorption were determined and the difference between the intercepts at zero pressure was used to estimate hydrogen uptake. The CO chemisorption capacity of Ru was determined by temperature programmed surface reaction in D_2 following reduction at 498 K in pure D_2 for two hours. Further details concerning this procedure are discussed in the Results section. The BET surface area and pore size distribution for each catalyst was determined from N_2 isotherms at 77K.

H_2, CO and He were supplied from a gas flow manifold to a low dead volume quartz microreactor. UHP H_2 (Matheson Gas) or D_2 supplied by U.C.L.B.L. were further purified by passage through a Deoxo unit (Engelhard Industries) and water was removed by a molecular sieve 13X trap. UHP CO (99.999% pure, Matheson Gas) was passed through a glass bead trap at 573 K to remove iron carbonyls,

Table 1

Catalyst characteristics

Catalyst	Ru (wt %)	Dispersion (%)	BET (m^2/gm)	H uptake (μmoles/gm)	CO/H	pore dia (Å)
Ru/TiO_2(D)	3.3	15.5	47.5	51.1	1.3	213
Ru/TiO_2(A1)	3.7	30.9	105.4	112.5	0.8	94
Ru/TiO_2(A2)	3.1	40.6	76.5	125.3	0.8	120
Ru/TiO_2(R)	6.1	30.8	46.0	186.6	2.2	167

an Ascarite trap to remove CO_2, and finally a molecular sieve trap. UHP He was passed through a molecular sieve trap. 99% ^{13}CO (Isotec Inc) was used as supplied.

The products of CO hydrogenation were analysed by gas chromatography using a Perkin Elmer Sigma 3B gas chromatograph equipped with a FID detector. C_1-C_{14} hydrocarbons were separated on a fused silica capillary column (0.25 mm i.d. x 50m) coated with SE-54 (1 μm thick). Temperature programming of the column was as follows: 253 K for 5 min, 20 K/min upto 523 K, and then 523K for 10 min. Since the analysis of each loop requires 28.5 min, samples of reaction products were taken as needed, stored at 393 K, and then introduced to the chromatograph after reaction. To accomplish this, the effluent from the reactor was routed through a 10-port, two-position valve (used either to sample the products or inject the stored sample into the chromatograph) in series with a 16 sample-loop multiposition valve (used to acquire and store the samples). Both valves were contained in heated ovens maintained at 393 K to avoid condensation of products. Operation of both valves was by motor-driven actuators connected to a personal computer (IBM-PC). The microcomputer was programmed to schedule sample acquisition and analysis at desired times. The output from the FID detector was acquired by the computer via an interface and stored at a rate of 2 data points per second.

During a temperature-programmed surface reaction experiment, the reactor effluent was monitored by a mass spectrometer (UTI 100C). The following masses were monitored: 15

($^{12}CH_4$), 21 ($^{13}CD4$), 28 (^{12}CO), 29 (^{13}CO), 44 ($^{12}CO_2$) and 45 ($^{13}CO_2$). Mass spectrometer control and data acquisition were carried out by the personal computer. Using this system each mass intensity could be sampled twice a second.

For each experiment, 0.2 to 0.3 gm of catalyst was loaded in the reactor. The catalyst was then reduced by slowly ramping the temperature in D_2 at 10 K/min to 503 K, and then holding at this temperature for at least two hours. The synthesis of hydrocarbons was carried out at 1 atm using ^{12}CO and D_2. All reactions were performed at the following conditions: T = 498K; $D_2/CO = 3$; $P_{CO} = 50$ torr; $P_{D2} = 150$ torr; $P_{He} = 560$ torr; Q = 150 cc/min. CO conversions on all the catalysts were < 10 %. Activity was followed as a function of reaction time (t_r) by sampling at different times after the start-up of the reaction, and data were collected up to four hours onstream. After reaction, the catalyst was reduced at 503 K overnight.

TPSR experiments were initiated by pretreating the catalyst in one of several ways, as discussed in the Results section. The reactor was flushed with He to remove gas phase reactants and the catalyst was quenched in He to room temperature in under one minute by blowing air over the outside of the reactor. If the uptake of CO was to be determined, the catalyst was exposed to 50 torr of ^{13}CO in He for 5 min, and then flushed in pure He. The final step in each TPSR experiment was to ramp the catalyst temperature at 0.17 K/s to 503 K in flowing D_2 (50 torr). During this period, the evolution of ^{13}C labeled products (including ^{13}CO and $^{13}CD_4$) was

monitored by the mass spectrometer.

RESULTS

Figure 1 shows a plot of the turnover frequency for CO conversion to hydrocarbons, N_{CO}, as a function of time under reaction conditions, for each of the four catalysts. N_{CO} was determined by summing the rate of formation of C_1 through C_{14} hydrocarbons weighted by the carbon number and dividing this sum by the moles of surface Ru atoms determined by hydrogen chemisorption. Each of the catalysts exhibits a rapid initial loss in activity which is then followed by a much slower decline. The initial activity could be restored by D_2 reduction at 498 K and the plots of N_{CO} versus time could be reproduced over several cycles of reduction and reaction. It was also observed that the room temperature uptake of CO after reduction of the partially deactivated catalysts was identical to that prior to reaction.

The initial activity at t_r = 1 min for $Ru/TiO_2(D)$, $Ru/TiO_2(A1)$ and $Ru/TiO_2(A2)$ exhibits an inverse linear correlation with Ru dispersion, as shown in Fig. 2. The point for $Ru/TiO_2(R)$ lies well below the correlation. For $t_r > 20$ min, the loss in activity can be fit by an exponential function, $N_{CO}=N_{CO}^0\exp(-t_r/\tau)$, where N_{CO}^0 is the apparent initial activity at t_r = 0 min and τ is the time constant for deactivation. The values of τ are 310 min for $Ru/TiO_2(D)$, 420 min for $Ru/TiO_2(A1)$, 500 min for $Ru/TiO_2(A2)$ and 714 min for

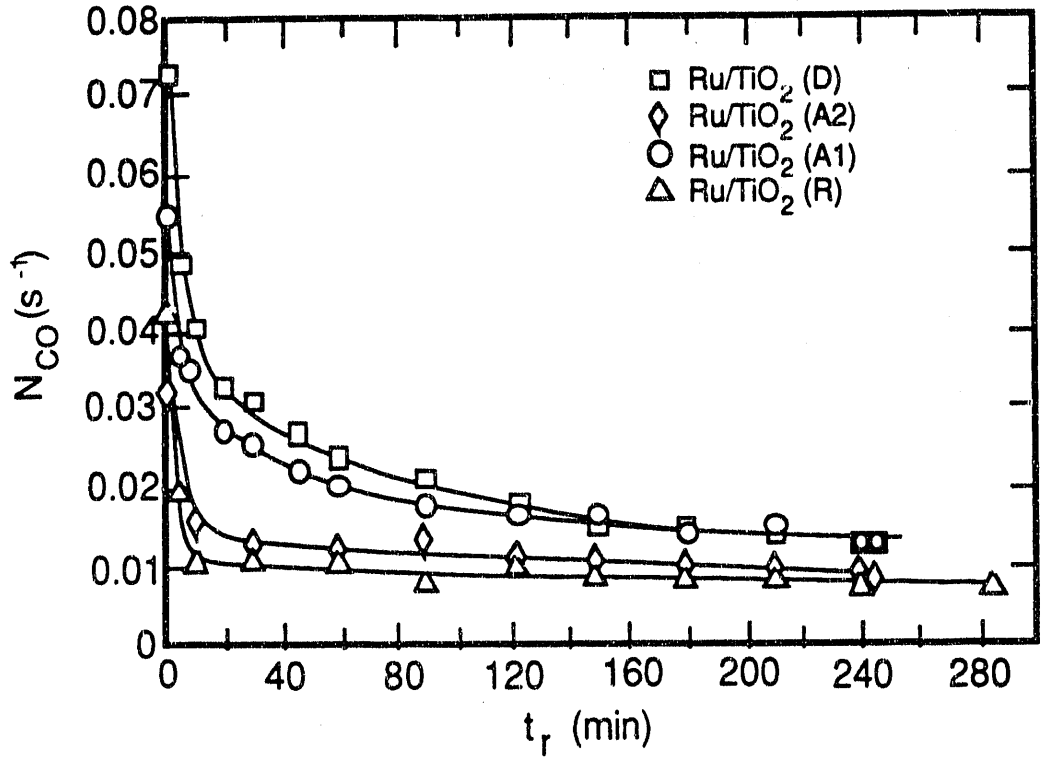

Fig. 1 CO turnover frequency as a function of reaction time. Reaction Conditions: P_{D2} = 150 torr, P_{CO} = 50 torr, T = 498 K.

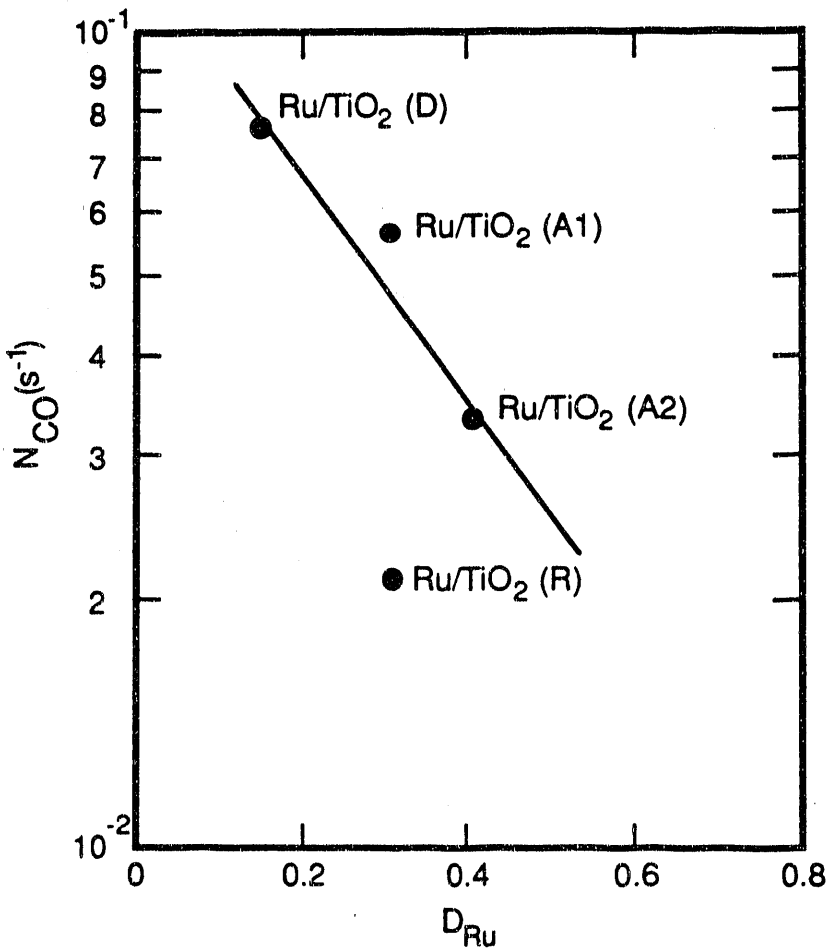

Fig. 2 Catalyst activity at 1 min after reaction start up versus ruthenium dispersion.

$Ru/TiO_2(R)$. Figure 3 shows that a linear correlation exists between τ^{-1} and N_{CO}^0, which suggests that whetever causes the slow deactivation of the catalyst is produced by the catalyst itself.

Figure 4 shows an Anderson-Schulz-Flory (23) plot of the distribution of C_1 through C_{14} hydrocarbons for t_r = 5 min and t_r = 120 min for $Ru/TiO_2(D)$. It is evident that the carbon number distribution of products is not strongly affected by deactivation. Similar observations were also made for the other three catalysts. Moreover, for all four catalysts, the probability of chain growth, α, was found to be 0.7± 0.01, indicating that α is independent of Ru dispersion and support properties (i.e., TiO_2 phase and surface area). The effects of reaction time on the proportions of total olefins, internal olefins, and branched products was examined at each carbon number between 2 and 7. For a given carbon number, the fraction of α-olefins was observed to rise during the first 10 min, at the same time that the fraction of β-olefins declined by an equivalent amount; however, neither the proportion of total olefins nor the proportion of branched products changed significantly with reaction time.

The CO adsorption capacity of each catalyst was measured both prior to the onset of reaction and after progressively longer periods of time under reaction conditions. The CO uptake of a freshly reduced catalyst was determined by exposing it to ^{13}CO at room temperature for 5 min, flushing the catalyst with He for 15 min and then carrying out a temperature-programmed surface reaction in D_2. A peak of $^{13}CD_4$ was observed. To estimate the CO

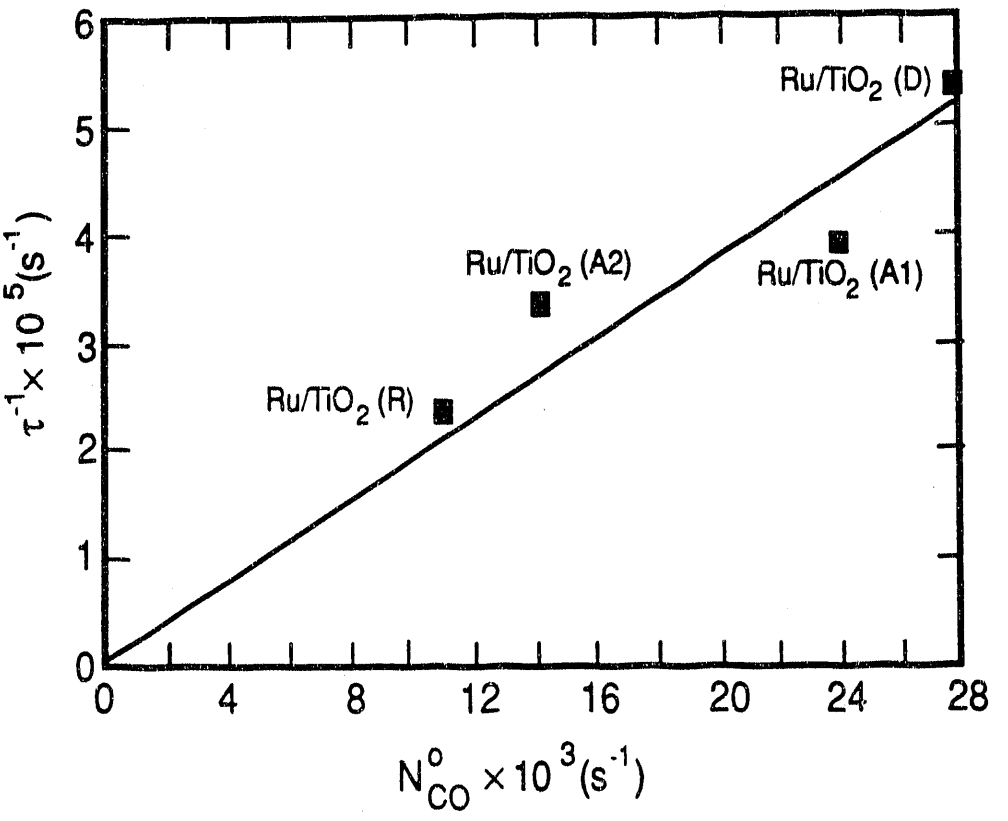

Fig. 3 Correlation of the inverse of the deactivation time constant with the CO turnover frequency at $t_r = 0$ obtained by extrapolation of the N_{CO} versus t_r curve for $t_r \geq 20$ min.

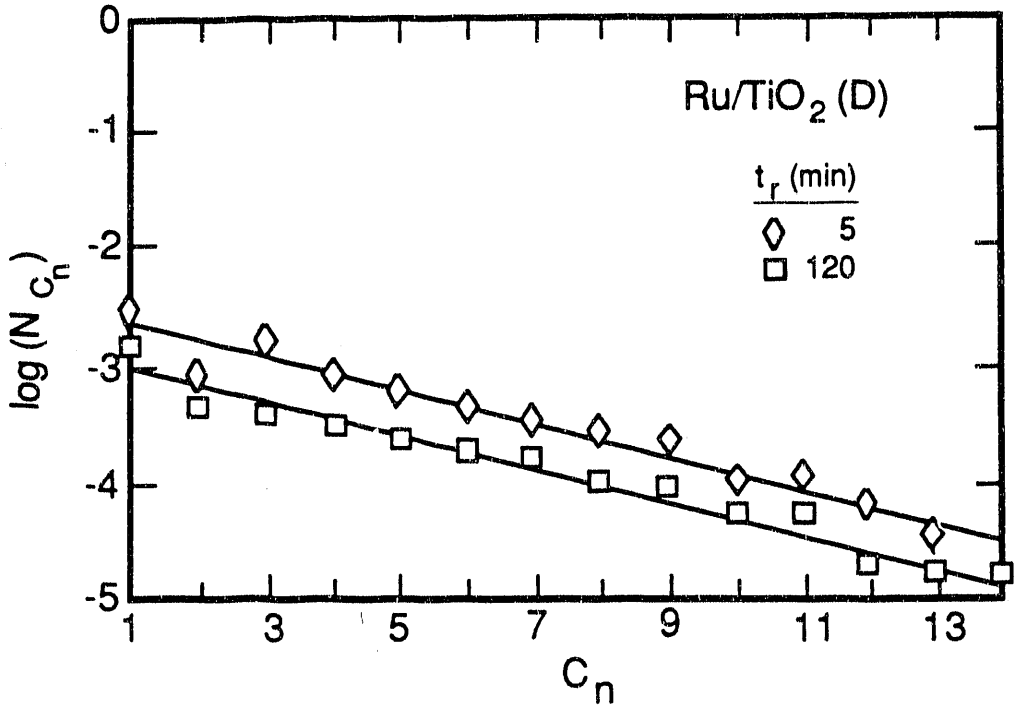

Fig. 4 Effect of reaction time on the ASF plot (log N_{Cn} vs C_n) for $Ru/TiO_2(D)$. N_{Cn} is the turnover frequency for products of carbon number n.

adsorption capacity after reaction, reaction was carried out in a mixture of $^{12}CO/D_2$ for a specified time after which the catalyst was cooled abruptly to room temperature in flowing He. The ^{12}CO remaining on the catalyst surface was displaced by exposing the catalyst to ^{13}CO for 5 min. The catalyst was then flushed in He for 15 min and a temperature-programmed surface reaction was carried out in flowing D_2.

The only ^{13}C-labeled product observed during the temperature-programmed surface reaction of adsorbed ^{13}CO was $^{13}CD_4$. As can be seen in Fig. 5, the amount of $^{13}CD_4$ formed decreases with increasing t_r. This change is accompanied by a progressive upscale shift in the temperature of the maximum rate of $^{13}CD_4$ formation. It is also apparent, particularly for $Ru/TiO_2(R)$, that the $^{13}CD_4$ spectrum for $t_r > 0$ consists of two or more overlapping peaks. The apparent activation energy for the reduction of adsorbed ^{13}CO was estimated from a plot of the log of the initial rate of $^{13}CD_4$ formation versus inverse temperature. A value of 30 ± 2.5 kcal/mol was obtained for all of the catalysts, independent of t_r.

The area under the $^{13}CD_4$ peaks obtained during temperature-programmed surface reaction of ^{13}CO is taken as the measure of the CO adsorption capacity of Ru. Figure 6 shows that as t_r increases, the CO adsorption capacity of each of the four catalysts decreases rapidly and then declines more slowly for $t_r \geq 5$ min. As can be seen from Table 2, the time constants for the slow loss of CO adsorption capacity are virtually identical to those for the slow loss in CO hydrogenation activity.

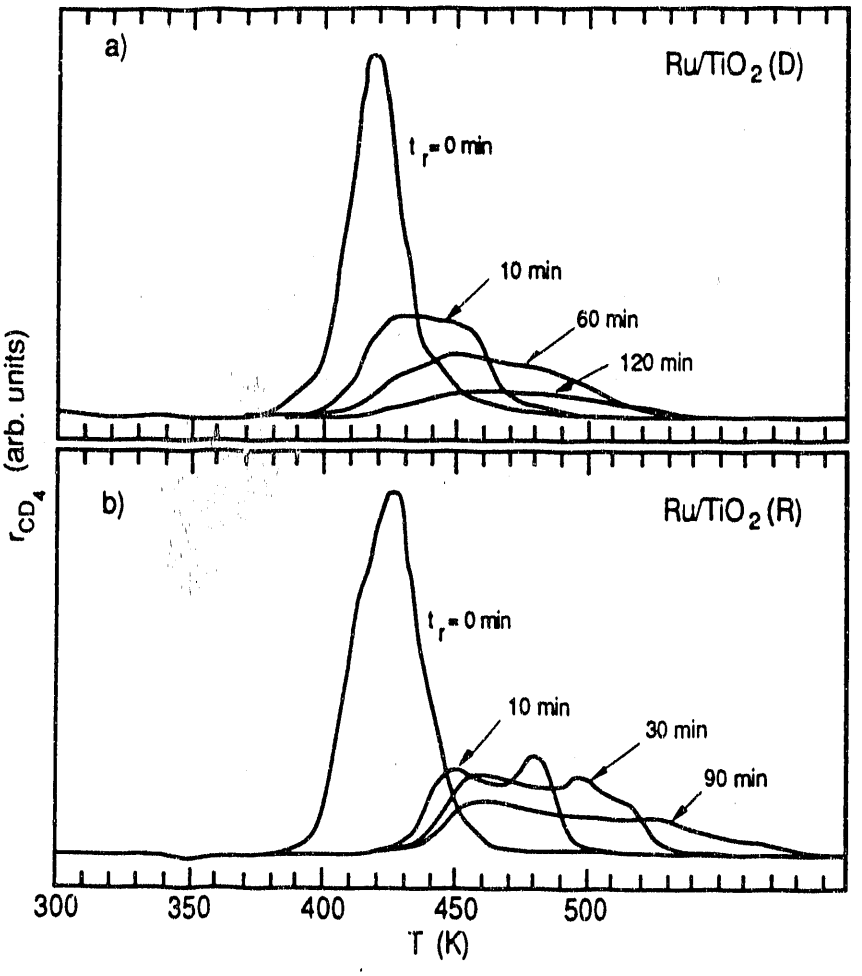

Fig. 5 ^{13}C TPSR spectra to estimate CO uptake as a function of t_r. a. Data for $Ru/TiO_2(D)$. b. Data for $Ru/TiO_2(R)$. Reaction Conditions: P_{D2} = 150 torr, P_{CO} = 50 torr, T = 498 K. Gas Introduction sequence: $^{12}CO + D_2$ (t_r) → He (cool) → ^{13}CO (5 min) → D_2 (TPSR). In the TPSR spectra, the temperature is raised from room temperature to 503 K at 0.17 K/sec and then maintained constant at 503 K.

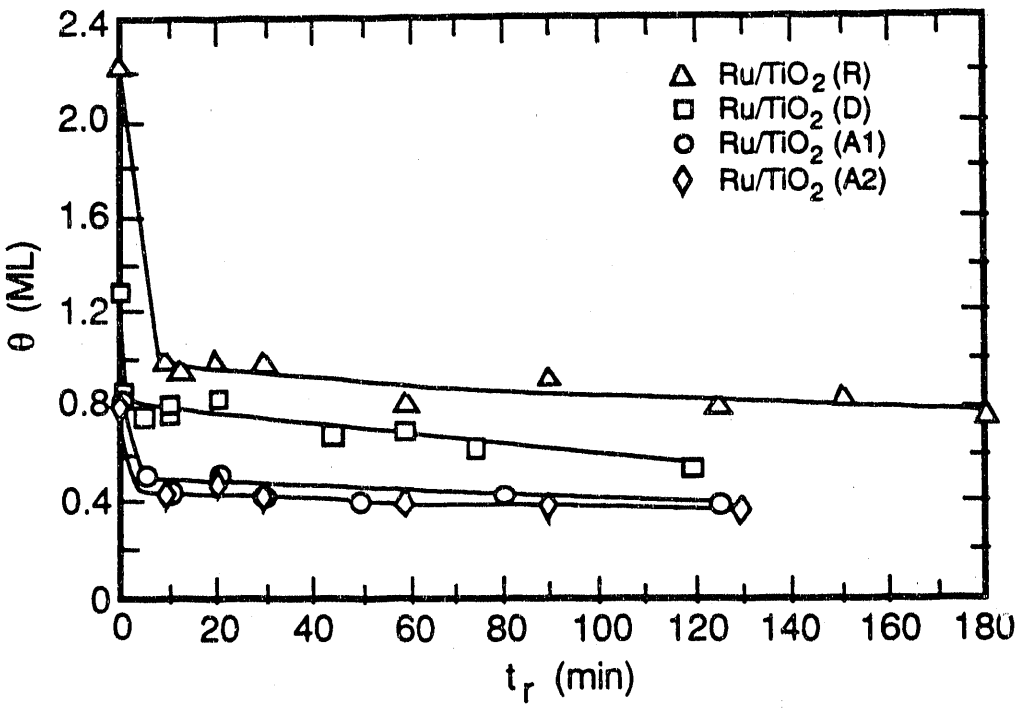

Fig. 6 CO uptake as a function of reaction time, t_r.

Table 2

Comparison of the time constants for the decay in N_{CO} with that for the decay in θ_{CO}, for $t_r > 20$ min

Catalyst	N_{CO} decay (min)	θ_{CO} decay (min)
$Ru/TiO_2(D)$	310	306
$Ru/TiO_2(A1)$	420	476
$Ru/TiO_2(A2)$	500	550
$Ru/TiO_2(R)$	714	715

Additional experiments were performed to determine the cause of the immediate loss of CO chemisorption capacity upon initiation of CO hydrogenation. The freshly reduced catalyst was exposed to ^{13}CO at room temperature for 5 min, flushed in He for 5 min, and then contacted with ^{12}CO in He for 5 min to exchange off all the adsorbed ^{13}CO. The catalyst was then flushed with He and TPSR in D_2/He was initiated. A pulse of $^{13}CO_2$ was observed when the catalyst contacted with ^{13}CO, and a peak of $^{13}CD_4$ was observed during TPSR. Both of these observations suggest that some of the adsorbed ^{13}CO disproportionates via the reaction: $2CO_s \rightarrow C_s + CO_2(g)$. As shown in Table 3, the amount of $^{13}C_s$ deposited by CO disproportionation correlates well with the immediate loss of CO chemisorption capacity observed upon initiation of CO hydrogenation over $Ru/TiO_2(D)$, $Ru/TiO_2(A1)$, and $Ru/TiO_2(A2)$. Similar results have been reported by Yamasaki *et al.* (7) who observed that CO disproportionation caused a reduction in the extent of CO adsorption. For $Ru/TiO_2(R)$, the loss in CO chemisorption capacity is much greater than the amount of carbon deposited by room-temperature disproportionation of CO. It is also observed that the amount of carbon deposited on $Ru/TiO_2(R)$ by CO disproportionation is significantly less than that deposited on the other three catalysts.

The accumulation of carbonaceous species on the catalyst was determined using the following procedure. Reaction was run for a specified duration using a feed mixture of ^{13}CO and D_2. The catalyst was then exposed for 30 s to ^{12}CO in He, to exchange the adsorbed ^{13}CO for ^{12}CO (13), after which the catalyst was rapidly cooled to

Table 3

Comparison of the initial Loss in θ_{CO} with that due to room temperature disproportionation of CO

Catalyst	Loss on running reaction (ML)	Room Temp. disprop. CO (ML)
$Ru/TiO_2(D)$	0.45	0.47
$Ru/TiO_2(A1)$	0.33	0.27
$Ru/TiO_2(A2)$	0.35	0.34
$Ru/TiO_2(R)$	1.23	0.13

room temperature in flowing He. Temperature-programmed surface reaction was then carried out in flowing D_2 and the formation of $^{13}CD_4$ was monitored as a function of time. Spectrum b in Fig. 7 shows the rate of $^{13}CD_4$ formation versus the temperature for the case of $Ru/TiO_2(D)$. Spectrum a shows the rate of $^{13}CD_4$ formation if the exchange of adsorbed ^{13}CO for ^{12}CO is eliminated from the procedure. The area under spectrum b corresponds to the total amount of carbonaceous species present on the catalyst, namely, the sum of C_α and C_β, whereas the area under spectrum a also includes the amount of chemisorbed CO.

As discussed by Yokomizo et al. (14), the C_β pool of carbon can be subdivided into two pools: a C_β' pool corresponding to alkyl precursors to C_{2+} products and a C_β'' pool corresponding to alkyl groups produced by reaction, but not associated with the formation of C_{2+} products. The amount of C_β'' was determined using the following procedure. The reaction was run in a mixture of ^{13}CO and D_2 for a time t_r. The feed was then switched to a mixture of ^{12}CO and D_2 for 2 min. Experiments performed to determine the reaction time required for the latter step showed that within 2 min, the adsorbed ^{13}CO would be displaced by ^{12}CO and all the 13-labeled carbon in the C_α and C_β' pools would be replaced by ^{12}C, but that the ^{13}C carbon in the C_β'' pool would remain. After reaction in the $^{12}CO/D_2$ mixture, the catalyst was rapidly cooled to room temperature in He, and a temperature-programmed surface reaction

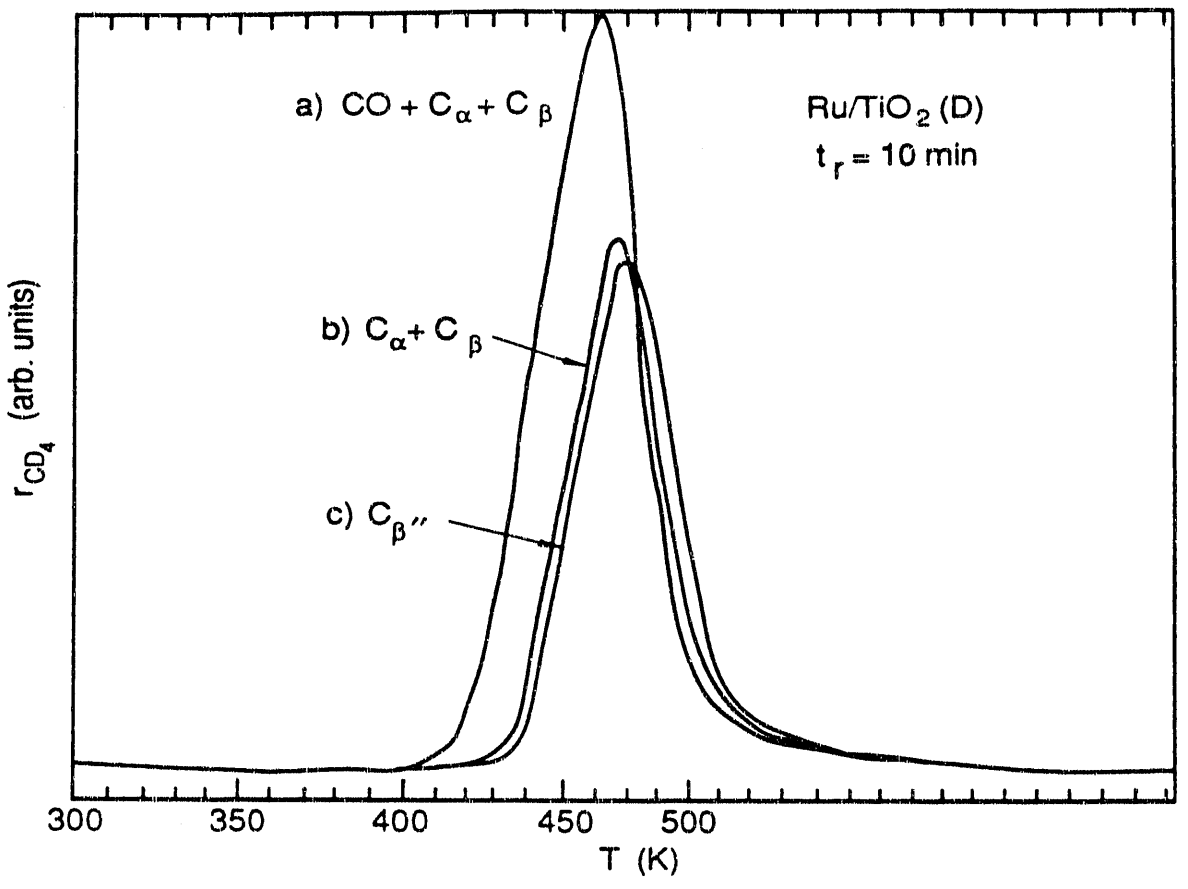

Fig. 7 ^{13}C TPSR spectra to estimate carbon accumulation on the catalyst at t_r = 10 min, standard reaction conditions. a. CO + C_α + C_β. Gas Introduction sequence: ^{13}CO + D_2 (10 min) → He (cool) → D_2 (TPSR). b. C_α + C_β: Gas Introduction sequence: ^{13}CO + D_2 (10 min) → ^{12} CO + He (30 s) → He (cool) D_2 (TPSR). c. C_β". Gas Introduction sequence: ^{13}CO + D_2 (10 min) → ^{12}CO + D_2 (2 min) → He (cool) → D_2 (TPSR). In the TPSR spectra, the temperature is raised from room temperature to 503 K at 0.17 K/sec and then maintained constant at 503 K.

was then carried out in flowing D_2. Spectrum c in Fig. 7 shows the resulting trace of $^{13}CD_4$ formation as a function of temperature. The small difference in the areas under spectra c and b represents the amount of C_α+C_β' carbon accumulated on the catalyst. Figure 8 shows that the amount of C_β" builds up monotonically as t_r increases.

The accumulation of C_α, C_β', and C_β" during the first 20 min of reaction are shown in Fig. 9. It is evident that the surface concentration of C_α and C_β' is less than 1 ML in most cases. For Ru/TiO_2(R), $RuTiO_2$(A1) and Ru/TiO_2(D), the accumulation of these forms of carbon passes through a maximum as t_r increases. Much larger quantities of C_β" are formed on the catalyst surface and these species increase in quantity monotonically, and after 20 min of reaction can be as high as 7 ML equivalents. The time constant for the build up of C_α+C_β during the first 10 min of reaction is listed in Table 4 and is compared with the time constant for the loss in CO hydrogenation activity during the same period. It is apparent that the time constants for the two phenomena are virtually identical, suggesting that the loss in CO hydrogenation activity during the initial stages of reaction can be attributed to the rapid buildup of C_α and C_β on the catalyst surface.

DISCUSSION

Previous investigations have shown that the specific activity

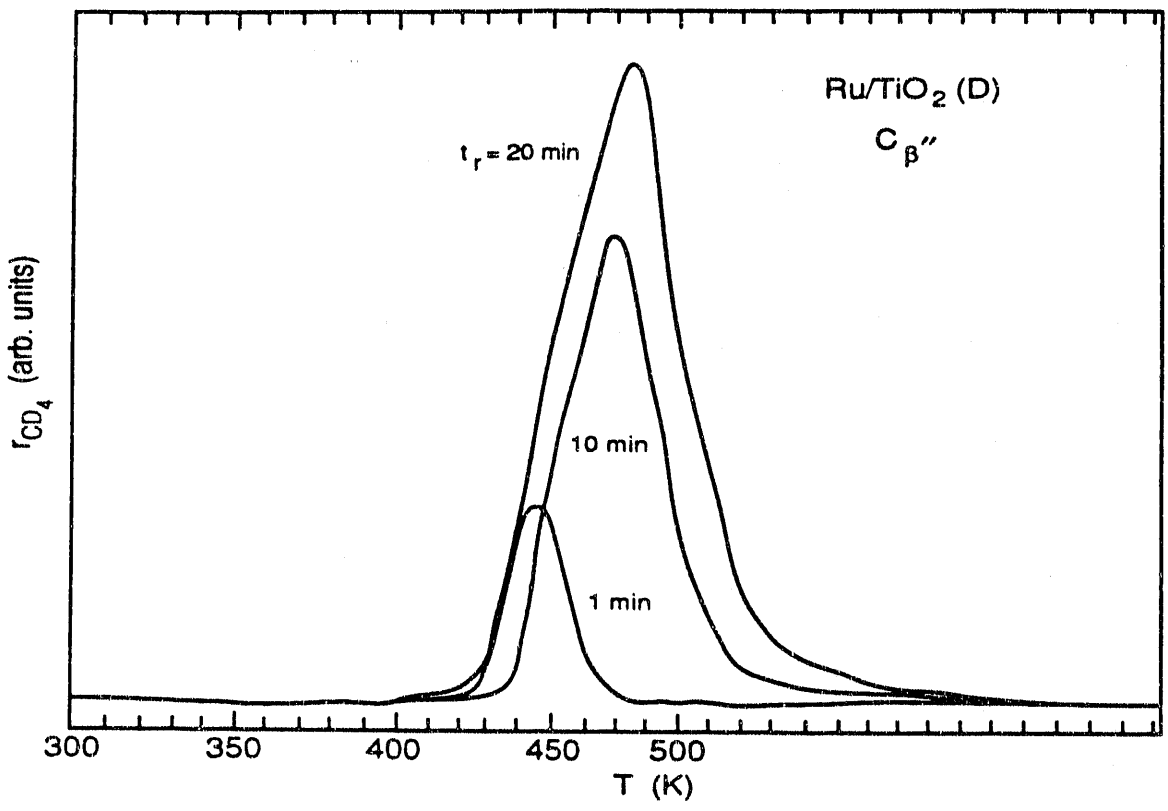

Fig. 8 ^{13}C TPSR spectra for C_β" with varying reaction time t_r. Gas introduction sequence: $^{13}CO + D_2$ (t_r) → $^{12}CO + D_2$ (2 min) → He(cool) → D_2 (TPSR). In the TPSR spectra, the temperature is raised from room temperature to 503 K at 0.17 K/sec and then maintained constant at 503 K.

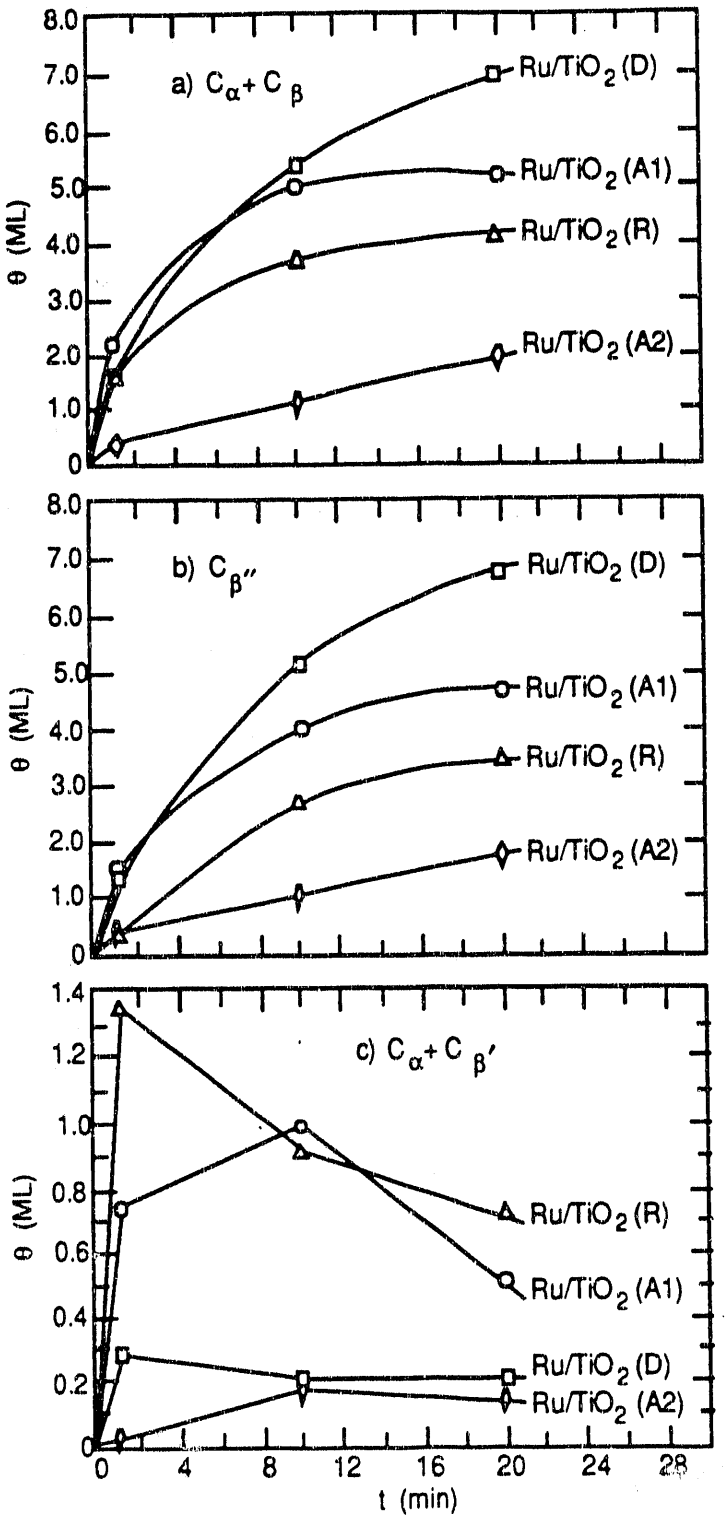

Fig. 9 Coverages of carbon species obtained from ^{13}C TPSR as a function of reaction time. a. $C_\alpha + C_\beta$. b. $C_{\beta''}$. c. $C_\alpha + C_{\beta'}$.

Table 4

Comparison of the time constants for the accumulation of $C_\alpha + C_\beta$ with that for the decay in N_{CO} during the first 10 min under reaction conditions

Catalyst	$(C_\alpha+C_\beta)$ (min)	N_{CO} (min)
$Ru/TiO_2(D)$	6.1	6.3
$Ru/TiO_2(A1)$	8.3	8.7
$Ru/TiO_2(A2)$	6.2	7.8
$Ru/TiO_2(R)$	8.3	8.3

of Ru for Fischer-Tropsch synthesis decreases with increasing Ru dispersion (24, 25). A similar trend has been observed in the present study for the CO turnover frequency measured after 1 min and for the turnover frequency determined by extrapolation of the slowly deactivating portion of the activity versus time plots (see Fig. 1). As can be seen in Fig. 2, with the exception of the point for $Ru/TiO_2(R)$, all of the data fall along a single line, suggesting that the surface area of titania does not affect the specific activity of Ru. The deviation of the point for $Ru/TiO_2(R)$ is believed to be due to the higher chloride content of this catalyst relative to the other three. Cl impurities can inhibit the catalytic activity of Ru for CO hydrogenation (26). Iyagba et al. (26) have shown that for a fixed Ru dispersion, the turnover frequency for CO hydrogenation decreases with increasing initial levels of chloride impurity. Based on their results, it is estimated that the Cl impurity in the $Ru/TiO_2(R)$ catalyst could be expected to reduce the activity of this catalyst byabout a factor of 2. This is nearly the same as the ratio of turnover frequencies for $Ru/TiO_2(A1)$ and $Ru/TiO_2(R)$, which have identical dispersions.

To determine whether the decrease in turnover frequency with increasing Ru dispersion is dependent on the support composition, a comparison was made between Ru/TiO_2 and Ru/Al_2O_3 catalysts. Figure 10 shows the turnover frequencies for methane obtained in the present study and similar results obtained by Kellner and Bell (25) using Ru/Al_2O_3 catalysts. It is evident that the slopes of the two sets of data are identical, indicating that the decrease in

turnover frequency with increasing dispersion is an intrinsic function of the Ru particle size, independent of the support composition. Figure 10 shows as well that at a fixed Ru dispersion, the methane turnover frequency is 2 times higher for titania-supported Ru than for alumina-supported Ru. Since the weight ratio of C_{2+} products to methane is roughly a factor of five higher for the titania supported catalysts, this means that the CO turnover frequency for the titania-supported catalysts is approximately a factor of ten higher than that for alumina-supported Ru.

The higher activity of TiO_2-supported Ru to Al_2O_3-supported Ru can be attributed to effects of highly active sites located at the boundaries of TiO_x islands located on the Ru metal surface. XPS studies (27) have suggested that the partial reduction of TiO_2 produces Ti^{3+} sites which can then interact with the oxygen of CO adsorbed on metal atoms immediately adjacent to the Ti^{3+} ions. This mode of CO adsorption has been proposed to weaken the C-O bond, and hence facilitate its cleavage, an important first step in CO hydrogenation (28-34). The TiO_x coverage is independent of Ru dispersion.

Kellner and Bell (25) have attributed the decrease in specific activity with increasing Ru dispersion to a geometric effect. They note that an essential step in the hydrogenation of CO to hydrocarbons is the dissociation of adsorbed CO. Indirect evidence suggests that this process occurs preferentially on flat metal surfaces, rather than on low coordination sites occurring at the corners and edges of metal particles (35, 36). Since the proportion

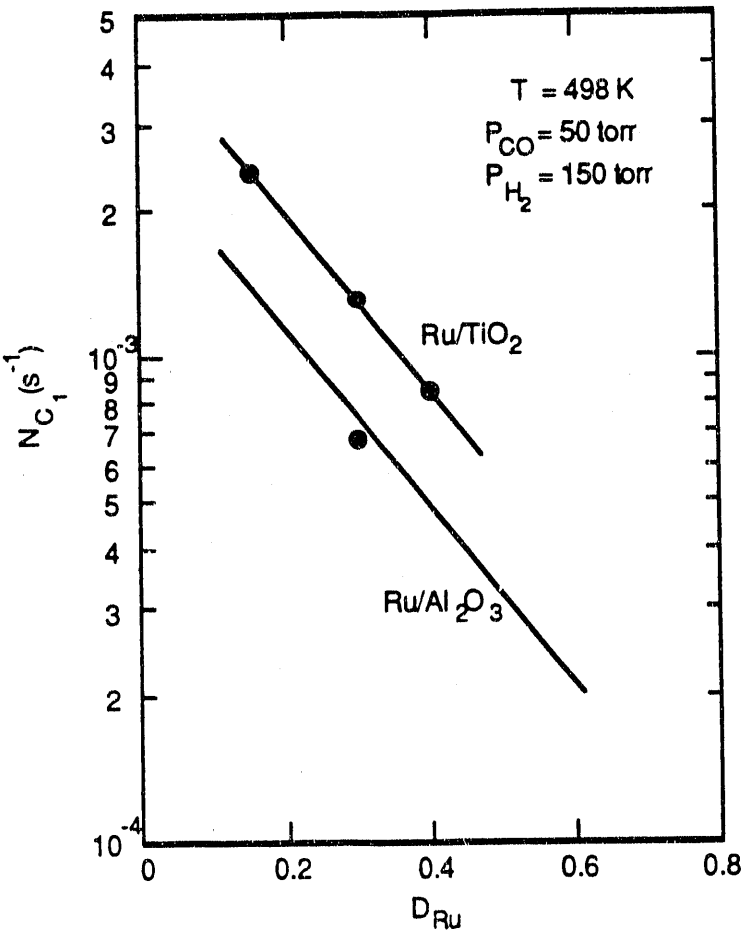

Fig. 10 Comparison of methane turnover frequency from this work over Ru/TiO_2 and Ru/Al_2O_3 (25). All data are based on measurements made using a H_2/CO feed mixture.

of surface metal sites occurring on flat planes decreases with increasing metal dispersion, it is possible to explain the results shown in Fig. 10. The postulation of CO dissociation as a rate-determining step in CO hydrogenation is also consistent with the observed distribution of products. As noted in the preceding section, the probability of chain growth, α, is independent of Ru dispersion. This together with the absence of any effect of dispersion on the olefin to paraffin ratio or the extent of chain branching suggest that the only step in the reaction sequence affected by Ru particle size is the initial process of CO dissociation.

The importance of CO dissociation in CO hydrogenation is also borne out by the observation that the room-temperature disproportionation of CO is most extensive on the catalyst exhibiting the highest initial activity, $Ru/TiO_2(D)$, and is least extensive on the catalyst with the lowest initial activity, $Ru/TiO_2(R)$ (see Table 3). Furthermore, the observation of room-temperature CO disproportionation on Ru/TiO_2 indicates a higher activity for these catalysts compared to silica-supported Ni, Co, and Ru, which require temperatures of at least 423 K (37).

All four Ru/TiO_2 catalysts exhibit very similar deactivation behavior with time under reaction conditions. There is a very rapid loss of activity during the first 20 min or so, whereafter the activity declines much more slowly. Deactivation does not change the product carbon number distribution; the product selectivity is independent of the degree of deactivation, the dispersion of Ru, and the phase of the titania support. Comparison of the data in Figs. 1

and 6 shows that the loss in synthesis activity is accompanied by a loss in CO chemisorption capacity. Hydrogen reduction of the catalyst fully restores the initial activity of the catalysts as well as their CO chemisorption capacity. What this indicates is that the loss in activity is reversible and does not appear to be associated with a decrease in metal dispersion.

For very short reaction times ($t_r \leq 1$ min), the loss in Ru activity can be attributed to the accumulation of C_α. As noted earlier, this species is a precursor to the formation of hydrocarbons. However, the increase in coverage of C_α will result in a loss in CO chemisorption capacity. This is also shown in Table 3. Hoffmann and Robbins (38) have also observed a loss in CO chemisorption capacity during CO hydrogenation over Ru (001) due to the formation of surface carbon. Figure 9c shows, though, that the coverage of C_α + C_β' rapidly passes through a maximum with reaction time. For reaction times longer than a few minutes, the coverage by C_β" becomes significant, and it is probably this species which is primarily responsible for the loss in catalytic activity and CO chemisorption capacity at longer reaction times. The TPSR spectra presented in Fig. 8 indicate that C_β does not undergo hydrogenolysis until much of the chemisorbed CO has been removed from the surface, in agreement with earlier observations by Kobori et al. (39) and Zhou and Gulari (16). It has been proposed (16, 40, 41) that the stability of alkyl groups in the C_β" pool is due to the stabilization of CO adsorbed on the same, or nearby, sites. It is interesting to note

in this context that White and coworkers (40) have reported a similar stabilization of ethylidyne by adsorbed CO on a Ni(111) surface.

The loss in catalytic activity for longer reaction times is attributed to the progressive accumulation of C_α and C_β. This explanation is suggested by the similarity in the time constants for the accumulation of C_α and C_β, and the time constant for deactivation during the first 10 min of reaction (see Table 4). While the results of the present study do not provide direct evidence of a correlation between activity loss and the accumulation of carbon for reaction times in excess of 20 min, one can infer such a relationship from the observed correlation between the loss in CO chemisorption capacity and activity (see Table 2), since the loss in CO chemisorption capacity is ascribable to the accumulation of carbonaceous species.

Further support for the idea that the slow deactivation is due to species produced by the catalyst can be drawn from Fig. 3. This figure shows that the rate of deactivation for $t_r > 20$ min increases linearly with the turnover frequency of the catalyst. For Ru/TiO_2(D), Ru/TiO_2(A1) and Ru/TiO_2(A2), the differences in the turnover frequency are due to differences in Ru dispersion (see Fig. 2). As discussed above, the lower than anticipated activity of Ru/TiO_2(R) is attributed to the higher initial Cl impurity of this catalyst relative to the other three investigated.

The observed pattern of deactivation and loss in CO chemisorption capacity can be represented qualitatively by the

following sequence of steps.

$$CO_g \overset{K}{\rightleftarrows} CO_s \overset{k_1}{\rightarrow} C_\alpha \overset{k_2}{\rightarrow} P$$
$$\downarrow k_3$$
$$C_\beta"$$

The first step is the reversible, nondissociative adsorption of CO. Isotopic tracer studies (8, 13, 42) have shown that this step can be assumed to be at equilibrium under reaction conditions. Dissociation of adsorbed CO produces C_α, which is viewed in the above scheme as a precursor to both hydrocarbon products and $C_\beta"$. The accumulation of $C_\beta"$ is assumed to reduce the fraction of metal sites available for CO adsorption and hence will inhibit the rate at which products are formed.

The general features of the proposed scheme are also supported by isotopic tracer studies conducted using Ru/SiO_2 (9, 11), Ru/Al_2O_3 (15), and Ru/TiO_2 (14). In as much as the work carried out with Ru/TiO_2 (14) has shown the surface coverage by C_β' to be small relative to C_α and $C_\beta"$, the presence of C_β' is not indicated explicitly in the proposed scheme, and in the analysis of the dynamics of carbon accumulation on the catalyst surface presented below, the coverage by C_β' is included implicitly with that by C_α. As written, the proposed scheme assumes that the coverage of the catalyst by hydrogen is time independent. In one tracer study, Wislow and Bell (10) observed close to monolayer coverages of both CO and D_2 on an unsupported Ru catalyst under reaction conditions,

indicating that CO and D_2 chemisorb on different sites.

Expressions for the rate of product formation and for the coverage of the catalyst surface by CO, C_α, and C_β" can be derived on the basis of the reaction scheme presented above. The rate of product formation can be expressed as

$$N_{CO} = k_2\theta_\alpha \qquad (1)$$

The coverage of the surface by CO, C_α, and C_β" is governed by the following relationships

$$\theta_{CO} = \frac{KP_{CO}(1-\theta_\alpha-\theta_\beta")}{1+KP_{CO}} \qquad (2)$$

$$\frac{d\theta_\alpha}{dt} = k_1\theta_{CO}-(k_2+k_3)\theta_\alpha \qquad (3)$$

$$\frac{d\theta_\beta"}{dt} = k_3\theta_\alpha \qquad (4)$$

where θ_{CO}, θ_α, and θ_β" are the coverages of CO, C_α, and C_β", respectively; P_{CO} is the partial pressure of CO; K is the equilibrium constant for CO adsorption; and k_i is the rate coefficient for reaction i. The initial conditions for eqns. 3 and 4 are $\theta_\alpha = \theta_\beta" = 0$.

Assuming that $KP_{CO} >> 1$, the time dependences of θ_α and θ_β" are given by

$$\theta_\alpha(t) = \frac{k_1(\exp(m_1 t)-\exp(m_2 t))}{\sqrt{(k_1+k_2+k_3)^2-4k_1k_3}} \qquad (5)$$

and

$$\theta_\beta''(t) = 1+ \frac{k_1k_3(\frac{\exp(m_1 t)}{m_1} - \frac{\exp(m_2 t)}{m_2})}{\sqrt{(k_1+k_2+k_3)^2-4k_1k_3}} \qquad (6)$$

where

$$m_1, m_2 = \frac{-(k_1+k_2+k_3)\pm\sqrt{(k_1+k_2+k_3)^2-4k_1k_3}}{2} \qquad \text{and } m_1>m_2$$

Substitution of eqns. 5 and 6 into eqn. 2 gives

$$\theta_{CO}(t) = \frac{k_1}{\sqrt{(k_1+k_2+k_3)^2-4k_1k_3}} (\exp(m_2 t)(1+\frac{k_3}{m_2})-\exp(m_1 t)(1+\frac{k_3}{m_1})) \qquad (7)$$

and

$$\frac{N_{CO}(t)}{\theta_{CO}(t)} = \frac{k_2(\exp(m_1 t)-\exp(m_2 t))}{(\exp(m_2 t)(1+\frac{k_3}{m_2})-\exp(m_1 t)(1+\frac{k_3}{m_1}))} \qquad (8)$$

Plots of N_{CO}/θ_{CO}, θ_{CO}, θ_α, and θ_β'' are shown in Fig. 11. The values of the three rate coefficients used to generate these plots were selected to be representative of the dynamics observed in the experiments reported here. Comparison of the curves shown in Fig. 11 with the data shown in Figs. 6 and 9 shows a number of similarities. The coverage of CO decreases very rapidly during the first two to three minutes and then decays more slowly. This

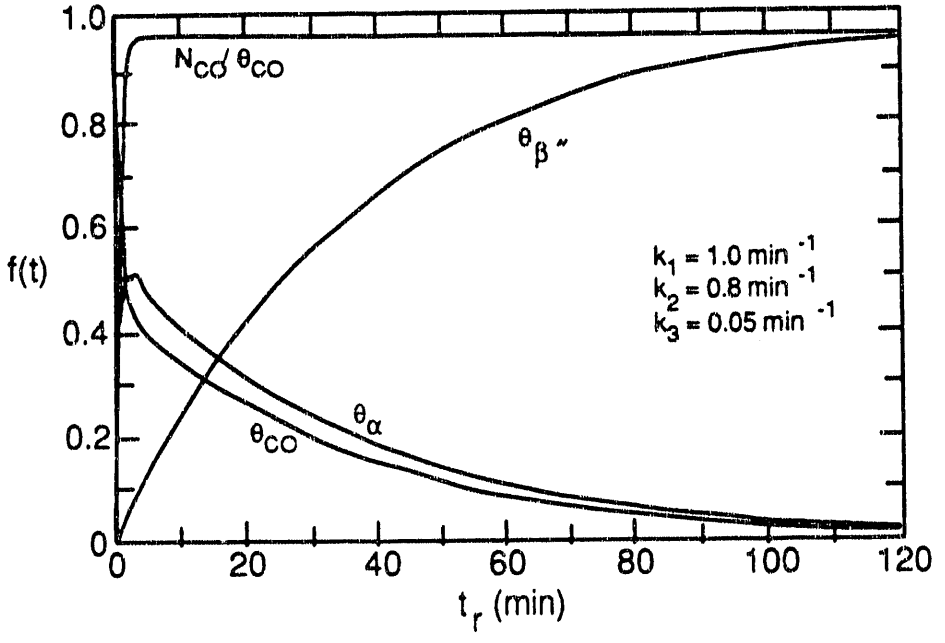

Fig. 11 Model curves for N_{CO}/θ_{CO}, θ_{α}, θ_{CO}, and θ_{β}" as a function of reaction time.

pattern can be attributed to the rapid initial build up of (C_α + C_β') and the subsequently slower build up of C_β". In agreement with experimental observation, the coverage by (C_α + C_β') passes through a maximum within the first few minutes of reaction, whereas the coverage by C_β" rises monotonically. The ratio N_{CO}/θ_{CO} rises rapidly to a fixed value which remains constant thereafter, in agreement with the observation that N_{CO} and θ_{CO} decline concurrently at $t_r \geq 10$ minutes.

The proposed model assumes that hydrogen chemisorption is not affected by the process of reaction and subsequent slow reduction in adsorbed CO on the catalyst surface. There is also an implicit assumption in the model that the chain length of the C_β" is time independent. Analysis of our data indicates that the ratio of the average chain length of the accumulated carbon at 10 min of reaction to that at 1 min after the start-up of reaction is 2.8 ± 0.6, while the ratio estimated between 20 min and 10 min is only 1.3 ±0.3. This indicates that the average chain length rises rapidly and then at a slower rate as reaction proceeds. A similar pattern has also been observed recently by Zhou and Gulari (16) for Ru/Al_2O_3. One can, therefore, envision a scenario in which the number of Ru surface sites blocked by each unit of C_β" decreases with time. During a period of rapidly increasing chain length (i. e., shortly after the start-up of reaction) the assumption of a constant chain length would lead to an overestimation of the effects of C_β" accumulation and, hence, to an underestimation in θ_{CO}. Despite these simplifying

assumptions, the model does qualitatively describe experimental observations.

Finally, one might ask whether the observed loss in activity might be due to the progressive accumulation of wax either on the surface of Ru particles or in the catalyst pores. To address this question, estimates were made of the wax accumulation after 2 h of reaction. The carbon number threshold for condensation of wax was determined using an extrapolation of the measured ASF distribution together with the Kelvin equation and known vapor pressures of alkanes. For the conditions of the present experiments, it was found that all products with 28 carbon atoms or more would condense in 10 Å pores, whereas all products with 83 carbon atoms or more would condense in 100 Å pores. The total wax accumulation could then be calculated for each catalyst using the measured pore size distribution. By this means, it was found that the wax could occupy between 0.06-0.55% of the pore volume in the four catalysts studied. If the wax is assumed to cover the Ru particles in a maximal fashion, then it is estimated that between 3% and 7% of the exposed Ru sites would be covered. These figures are to be compared with fractional activity losses of 23-46% over the same period of time. This suggests that wax accumulation cannot account for a significant part of the loss in activity under the conditions used in this study. Of further note is the fact that the estimated amount of wax accumulation is only 1-4% of the amount of carbon removed from the catalyst surface after reaction.

CONCLUSIONS

Titania-supported Ru catalysts undergo deactivation during CO hydrogenation, but exhibit no change in product selectivity with time onstream. The initial activity can be restored by reduction in H_2 or D_2 and there is no evidence of metal sintering due to reaction. The rate of deactivation is most rapid during the first 20 min after the startup of reaction and declines to a much smaller rate for reaction times longer than 20 min. The long-term rate of deactivation is proportional to the activity of the catalyst, obtained by extrapolation of the long-term activity data. The loss in activity is paralleled by a loss in CO chemisorption capacity, and for reaction times longer than 20 min, the ratio of the CO turnover frequency to the CO coverage remains constant. Differences in the initial activity of Ru supported on Degussa P-25 and anatase can be attributed to differences in Ru dispersion, with higher dispersion contributing to a decrease in CO turnover frequency. For the same dispersion, Ru supported on rutile exhibits a two to threefold lower activity than Ru supported on anatase. The lower activity of rutile-supported Ru is attributed to the higher level of chloride impurity on rutile versus anatase. The formation of highly active sites at the boundaries of TiO_x islands located on the metal surface is proposed to be the cause for the higher activity of TiO_2-supported Ru, compared to SiO_2- or Al_2O_3-supported Ru.

Temperature-programmed surface reaction experiments show evidence of the accumulation of carbidic, C_α, and alkyl, C_β, carbon on

the catalyst surface. The catalyst inventory of (C_α + C_β') increases, passes through a maximum, and then decreases as the catalyst loses activity. On the other hand, the inventory of C_β" increases monotonically with reaction time. The accumulation of both C_α and C_β carbon results in a reduction in the CO chemisorption capacity of the catalyst. The immediate loss in CO uptake after reaction start-up is attributed to the buildup of C_α. The rapid loss in activity in the first ten minutes correlates with the buildup of C_α and C_β, whereas the slower loss in activity is ascribed to the accumulation of C_β". The alkyl chains comprising the C_β pool of carbon do not undergo extensive hydrogenolysis under reaction conditions, very likely due to inaccessibility of the chains to hydrogen as a result of CO adsorbed on adjacent sites. A model has been proposed that qualitatively agrees with the experimental obseravtions of changes in activity, CO uptake, and the accumulation of reactive and unreactive carbon species on the catalyst surface as reaction proceeds.

REFERENCES

1. Dalla Betta, R. A., Piken, A. G., and Shelef, M., *J. Catal.* **35**, 54 (1974).
2. Dalla Betta, R. A., Piken, A. G., and Shelef, M., *J. Catal.* **40**, 173 (1975).
3. Everson, R. C., Woodburn, E. T., and Kirk, A. R. M., *J. Catal.* **53**,

186 (1978).

4. Ekerdt, J. G., and Bell, A. T., *J. Catal.* **58**,170 (1979).
5. Moeller, A. D., and Bartholomew, C. H., *Ind. Eng. Chem. Prod. Res. Dev.* **21**, 390 (1982).
6. Bowman, R. M., and Bartholomew, C. H., *Appl. Catal.* **7**,179 (1983).
7. Yamasaki, H., Kobori, Y., Naito, S., Onishi, T., and Tamaru, K., *J. Chem. Soc., Far. Trans. 1* **77**, 2913 (1981).
8. Cant, N. W., and Bell, A. T., *J. Catal.* **73**, 257 (1982).
9. Winslow, P., and Bell, A. T., *J. Catal.* **86**, 158 (1984).
10. Winslow, P., and Bell, A. T., *J. Catal.* **91**, 142 (1985).
11. Duncan, T. M., Winslow, P., and Bell, A. T., *J. Catal.* **93**, 1 (1985).
12. Duncan, T. M., Reimer, J. A., Winslow, P., and Bell, A. T., *J. Catal.* **93**, 305 (1985).
13. Winslow, P., and Bell, A. T., *J. Catal.* **94**, 385 (1985).
14. Yokomizo, G. H., and Bell, A. T., *J. Catal.* **119**, 467 (1989).
15. Zhou, X., and Gulari, E., *J. Catal.* **105**, 499 (1987).
16. Zhou, X., and Gulari, E., *Langmuir* **4**, 1132 (1988).
17. Kikuchi, E., Matsumoto, N., Takahashi, T., Machino, A., and Morita, Y., *Appl. Catal.* **10**, 251 (1984).
18. Parfitt, G. D., *Trans. Far. Soc.* **67**, 2469 (1971).
19. Tauster, S. J., Fung, S. C., and Garten, R. L., *J. Amer. Chem. Soc.* **100**, 170 (1978).
20. Dalla Betta, R. A., *J. Catal.* **34**, 57 (1974).
21. Taylor, K. C., *J. Catal.* **38**, 299 (1975).
22. Kubicka, H., *React. Kin. Cat. Lett.* **5**, 223 (1976).
23. Anderson, R. B., in *"The Fischer-Tropsch Synthesis,"* Academic

Press, New York, 1984.

24. King, D. L., *J. Catal.* **61**, 77 (1980).
25. Kellner, C. S., and Bell, A. T., *J. Catal.* **75**, 251 (1982).
26. Iyagba, E. J., Hoost, T. E., Nwalor, J. U., and Goodwin, J. G., *J. Catal.* **123**, 1 (1990).
27. Levin, M. E., Salmeron, M., Bell, A. T., and Somorjai, G. A., *Surf. Sci.* **195**, 429 (1988).
28. Bracey, J. D., and Burch, R., *J. Catal.* **86**, 384 (1984).
29. Anderson, J. B. F., Bracey, J. D., Burch, R., and Flambard, A. R., *"Proceedings, 8th International Congress on Catalysis,"* Vol V, p. 111. Berlin, 1984.
30. Sachtler, W. M. H., in *"Proceedings, 8th International Congress on Catalysis,"* Vol V, p.151. Berlin, 1984.
31. Sachtler, W. M. H., Shriver, D. F., Hollenberg, W. B., and Long, A. F., *J. Catal.* **92**, 429 (1985).
32. Vannice, M. A., and Sudhakar, C., *J. Phys. Chem.* **88**, 2429 (1984).
33. Rieck, J. S., and Bell, A. T., *J. Catal.* **99** 262 (1986).
34. Levin, M. E., Salmeron, M., Bell, A. T., and Somorjai, A. T., *J. Catal.* **106**, 401 (1987).
35. Boudart, M., and McDonald, M. A., *J. Phys. Chem.* **88**, 2185 (1984).
36. Rieck, J. S., and Bell, A. T., *J. Catal.* **103**, 46 (1987).
37. Rabo, J. A., Risch, A. P., and Poutsma, M. L., *J. Catal.* **53**, 295 (1978).
38. Hoffmann, F. M., and Robbins, J. L., in *"Proceedings of the 9th International Congress in Catalysis, Calgary, !989"*, Vol. 3,

p.1144. Calgary, 1989.

39. Kobori, Y., Yamasaki, H., Naito, S., Onishi, T., and Tamaru, K., *J. Chem. Soc., Far. Trans. I* **78**, 1473 (1982).

40. Akhter, S., and White, J. M., *Surf. Sci.* **180**, 19 (1987).

41. Akhter, S., Henderson., M. A., Mitchell, G. E., and White, J. M., *Langmuir* **4**, 246 (1988).

42. Biloen, P., Helle, J.N., van den Berg, F. G. A., and Sachtler, W. M. H., *J. Catal.* **81**, 450 (1983).

Chapter III

Intrinsic Rate Parameters of the Fischer-Tropsch Synthesis

ABSTRACT

Fischer-Tropsch synthesis results in a large range of products, from methane to higher hydrocarbons. Steady-state product distributions obtained during the Fischer-Tropsch synthesis can be described by the Anderson-Schulz-Flory (ASF) polymerization model. The product distribution is characterized by a single parameter, α, the probability of chain growth, which is the ratio of the apparent rate constant for propagation (k_p') to the sum of the apparent rate constant for propagation and the rate constant for termination (k_t). Expressions are derived for k_p', k_t, and the average hydrocarbon product carbon number. It is shown that steady state rate data cannot be used to evaluate k_p' and k_t individually.

The reaction mechanism of Fischer-Tropsch synthesis is viewed as a surface polymerization in which the monomer is produced *in situ* from CO and H_2 on the catalyst surface (1-3). The experimentally observed product distribution obeys Anderson-Schulz-Flory (ASF) polymerization kinetics: $N_{Cn}/N_{Cn-1} = \alpha$, where N_{Cn} is the turnover frequency of products of carbon number n and α is the chain growth parameter and is independent of chain length n. This is indicative of propagation by a sequence of independently repeated additions of a monomer group to the growing chain, in competition with the termination step.

$$C_{n-1}(s) \xrightarrow{k_p'} C_n(s) \xrightarrow{k_p'} C_{n+1}(s)$$
$$C_n(s) \xrightarrow{+C_1(s)} C_n(g)$$

The ASF distribution can be expressed by the following two equations:

$$N_{C_n} = k(1-\alpha)^2\alpha^{n-1} \qquad (1)$$

$$\alpha = \frac{k_p'}{k_p'+k_t} \qquad (2)$$

where k is a rate parameter and $k_p' = k_p\theta_m$, k_p being the intrinsic propagation rate and θ_m the coverage of the monomer building block. N_{Cn}, α and k can be determined from steady-state data. The objective is to derive individual expressions for k_p' and k_t.

N_{CO} can be expressed in terms of the turnover frequency for producing a product with n carbon atoms, N_{Cn}, in the following

manner.

$$N_{CO} = \sum_{n=1}^{\infty} nN_{C_n} \qquad (3)$$

Kellner and Bell (4) have shown that N_{Cn} can be expressed as

$$N_{C_n} = k_t\theta_n \qquad (4)$$

where θ_n is the surface coverage by chains containing n carbon atoms. If chain growth occurs by the addition of single carbon units, it then follows that

$$\theta_n = \theta_1\alpha^{n-1} \qquad (5)$$

Now, the average product carbon number, n_{av} is given by the following expression:

$$n_{av} = \frac{\sum_{n=1}^{\infty} nN_{C_n}}{\sum_{n=1}^{\infty} N_{C_n}} \qquad (6)$$

Substitution of eqns. 3 and 4 into eqn. 6 results in

$$n_{av} = \frac{N_{CO}}{k_t \sum_{n=1}^{\infty} \theta_n} \qquad (7)$$

By substitution of eqns. 7 and 8 into eqn. 9, it can be shown that

$$n_{av} = \frac{\sum_{n=1}^{\infty} n\alpha^{n-1}}{\sum_{n=1}^{\infty} \alpha^{n-1}} \qquad (8)$$

and for $\alpha < 1$, this simplifies to

$$n_{av} = \frac{1}{(1-\alpha)} \qquad (9)$$

Using eqns. 7 and 9 in conjunction with eqns. 1 and 2, it can further be shown that

$$k_t = \frac{k(1-\alpha)}{\sum_{n=1}^{\infty} \theta_n} \qquad (10)$$

and

$$k_p' = \frac{k\alpha}{\sum_{n=1}^{\infty} \theta_n} \qquad (11)$$

Evaluation of k_p' and k_t requires determination of the sum of the surface coverages of the precursors to the gas-phase hydrocarbon products. It is evident from the above analysis that steady state rate data cannot be used to evaluate k_p' and k_t despite claims to have done so (5,6), since such data do not provide the sum of surface coverages appearing in the denominator of eqns. 10 and 11. As has already been noted by Zhang and Biloen (7), k_p' and k_t can be properly evaluated from transient isotopic tracer experiments.

Acknowledgements

The equations in this chapter were derived jointly with G. H. Yokomizo.

REFERENCES

1. Bell, A. T., *Catal. Rev.-Sci. Eng.* **23**, 203 (1981).

2. Biloen, P., and Sachtler, W. M. H. , *Adv. Catal.* **30**, 165, 1981.

3. Anderson, R. B., 'The Fischer-Tropsch Synthesis', Academic Press, New York, 1984.
4. Kellner, C. S., and Bell, A. T., *J. Catal.* **70**, 216 (1981).
5. Fu, L., Rankin, J. L., and Bartholomew, C. H., *C_1 Mol. Chem.* **1**, 369 (1986).
6. Adesina, A. A., *J. Catal.* **124**, 297 (1990).
7. Zhang, X. and Biloen, P., *J. Catal.* **98**, 468 (1986).

Chapter IV

Estimates of the Rate Coefficients for Chain Initiation, Propagation, and Termination during Fischer-Tropsch Synthesis over Ru/TiO_2

ABSTRACT

Transient response isotopic tracer experiments have been used to study chain growth during Fischer-Tropsch synthesis over a Ru/TiO_2 catalyst. This involves observation of the incorporation of ^{13}C into reaction products after an abrupt switch from $^{12}CO/D_2$ to $^{13}CO/D_2$ in the feed. Values for the rate constants for initiation, propagation and termination are determined by fitting theoretically generated model curves to the observed transient responses. The rate constant for chain initiation is independent of temperature and D_2/CO ratio. The rate constants for propagation and termination increase with temperature. The rate constant for propagation is not affected by the D_2/CO ratio. The rate constant for termination increases linearly with increasing D_2/CO ratio. The activation energy for chain termination is significantly higher than that for chain propagation, explaining the observed decrease in chain growth probability, α, with increasing temperature. Coverages by reaction intermediates are also estimated. The dominant species are monomeric building units which occupy 0.2-0.6 ML. Alkyl species

that are the direct hydrocarbon product precursors occupy < 0.2 ML and adsorbed CO covers 0.7 ML.

1.0 INTRODUCTION

Fischer-Tropsch synthesis (FTS) produces a spectrum of products consisting primarily of linear olefins and alkanes (1). The carbon number distribution of the products containing four or more carbon atoms is often well described by an Anderson-Schulz-Flory (ASF) distribution, which assumes that products containing n carbon atoms are produced by a stepwise polymerization of C_1 species (2). If the probability of chain growth, α, defined as the ratio of the rate of chain propagation to the sum of the rates of chain propagation and termination, is taken to be independent of n, then $[N_{Cn+1}]/[N_{Cn}] = \alpha$, where N_{Cn} and N_{Cn-1} are the turnover frequencies for products containing n and n+1 carbon atoms, respectively. Steady state investigations of FTS have shown that α is a complex function of both temperature and the partial pressures of H_2 and CO (2-5).

Mechanistic studies of hydrocarbon production by FTS over Fe, Co and Ru suggest that the mechanism of FTS might be described by the following reaction sequence (6):

1. $CO_g \rightleftarrows CO_s$
2. $H_{2,g} \rightleftarrows 2H_s$
3. $CO_s \rightarrow C_s + O_s$
4. $O_s + H_s \rightarrow OH_s$
5. $H_s + H_s \rightarrow H_2O_s$
6. $C_s + H_s \rightleftarrows CH_s$
7. $CH_s + H_s \rightleftarrows CH_{2,s}$
8. $CH_{2,s} + H_s \rightleftarrows CH_{3,s}$
9. $CH_{3,s} + H_s \rightarrow CH_{4,g}$

10. $CH_{3,s} + CH_{2,s} \rightarrow CH_3CH_{2,s}$
11. $CH_3CH_{2,s} + H_s \rightarrow CH_3CH_{3,g}$
12. $CH_3CH_{2,s} \rightarrow CH_2CH_{2,g} + H_s$
13. $CH_3CH_{2,s} + CH_{2,s} \rightarrow CH_3CH_2CH_{2,s}$

. . .

14. $C_nH_{2n+1,s} + CH_{2,s} \rightarrow C_{n+1}H_{2(n+1)+1,s}$
15. $C_nH_{2n+1,s} + H_s \rightarrow C_nH_{2n+2,g}$
16. $C_nH_{2n+1,s} \rightarrow C_nH_{2n,g} + H_s$

Adsorbed CO covers most of the catalyst surface and is in equilibrium with gas-phase CO (6-8). Likewise, dissociatively adsorbed H_2 is found to be in equilibrium with gas-phase H_2 (8). Some of the adsorbed CO dissociates irreversibly (7, 9) to form C_s and O_s. The atomic oxygen produced in this manner reacts rapidly with H_s to form water, whereas the atomic carbon reacts with hydrogen to form CH_x (x = 1-3) species (9, 10). The $CH_{3,s}$ species can react with additional hydrogen to form CH_4 or react with a $CH_{2,s}$ species, thereby initiating the process of chain growth. Chain propagation is sustained by $CH_{2,s}$ addition to adsorbed alkyl species. Chain termination can occur by hydrogen addition to a surface alkyl species to give a paraffin or by hydrogen abstraction to give an olefin.

A number of investigators have used isotopic-tracer techniques to identify the species involved in the chain growth process and to estimate the rate coefficients for chain propagation and termination (7, 11-20). In studies conducted with a Ru/SiO_2 catalyst, Kobori *et al.* (11) concluded that hydrocarbons are formed

by the polymerization of CH_x intermediates, and that CO insertion plays no part in the chain growth process leading to hydrocarbons. Work by Biloen *et al.* (12) on unsupported Co, Ni/SiO_2 and $Ru/\gamma\text{-}Al_2O_3$ has shown that carbidic intermediates are involved in the reaction and that methane and higher hydrocarbons have a common precursor. Stockwell *et al.* (17) were unable to conclude whether chain growth occurred via a CO or a CH_x insertion mechanism. Studies on an Fe/Al_2O_3 catalyst by Stockwell *et al.* (18) implicated a CH species as responsible for chain growth. They concluded that once chain growth is initiated, methane and higher hydrocarbon production is rapid and hence, coverages by the growing chains is small.

Several attempts to determine the rate coefficients for chain propagation and termination have been reported. Zhang and Biloen (14) have looked at the successive incorporation of ^{13}C into $C_{1\text{-}3}$ hydrocarbon products over Co and Ru catalysts. For Co, it was estimated that at 483 K, the value of the apparent rate constant for propagation decreased from 0.045 to 0.025 s^{-1} and the corresponding value of the apparent rate constant for chain termination increased from 0.02-0.04 s^{-1}, as the D_2/CO ratio increased from 1 to 6.55. For the same reaction conditions, the coverage by alkyl chains was estimated to be between 0.19 and 0.25 ML, whereas the coverage by monomeric species was much smaller. For Ru, the apparent rate coefficient for chain propagation was estimated to be $\geq 1 s^{-1}$. Based on similar experiments, Mims *et al.* (15) have estimated the apparent rate coefficient for chain propagation to be 2-4 s^{-1} for a promoted Fe catalyst at 510 K and a H_2/CO ratio of 1 (15), and 2 s^{-1}

on Co at 475 K and a H_2/CO ratio of 2 (16). Over both Fe and Co, the coverage by the monomeric species exceeded the coverage by hydrocarbon chains. In a recent study using a Ru/TiO_2 catalyst, Yokomizo and Bell (21) observed that 80 % of the carbidic species on the Ru surface served as a monomer building block, and 20 % acted as the precursor to methane. The rate coefficient for chain termination was estimated to be 0.044 s^{-1} at T = 463 K, p_{CO} = 50 torr and a D_2/CO = 3. The total coverage by C_1 species was reported to be 0.25 ML, and the coverage by C_{2+} species leading to hydrocarbon product was estimated at 0.1 ML.

The objective of the present investigation is to estimate the rate coefficients for chain initiation, propagation and termination, and to study their dependence on temperature and D_2/CO ratio. A further aim of this effort is to determine the surface coverages by the various reactive carbonaceous species present on the catalyst. A Ru/TiO_2 catalyst was used for these investigations because of the high specific activity of Ru for FTS and its characteristically high value of α (22). The absence of alcohols from the products of FTS over Ru was a further reason for choosing this metal. Titania was used to support and disperse the ruthenium since titania-supported Group VIII metals are known to be more active than silica- or alumina-supported Group VIII metals (23-26).

2.0 EXPERIMENTAL

2.1 Catalyst Preparation and Characterization

A 3.3% Ru/TiO_2 catalyst was prepared by incipient wetness

impregnation of Degussa P-25 titania with an aqueous solution of $RuCl_3.H_2O$ (Strem). Details of the preparation are given in ref. (27). The dried catalyst was reduced at 503 K to minimize encapsulation of the Ru crystallites by titania. The metal content of the catalyst was determined by X-ray fluorescence. The chloride level after reduction was below the detection limit (0.02%). The dispersion of Ru was determined by H_2 chemisorption to be 15.5%, while the CO uptake on the freshly reduced catalyst was 1.3 ML (ML based on H_2 chemisorption). The BET surface area, determined from a N_2 isotherm at 77 K, was 47.5 m^2/gm.

2.2 Apparatus

D_2, ^{12}CO, ^{13}CO and He were supplied from a gas manifold to a low dead volume quartz microreactor. UHP H_2 (Matheson Gas) or D_2 (Union Carbide) were further purified by passage through a Deoxo unit (Engelhard Industries) and water was removed by a molecular sieve 13X trap. UHP CO (99.999% pure, Matheson Gas) was passed through a glass bead trap maintained at 573 K to remove iron carbonyls, an Ascarite trap to remove CO_2, and a molecular sieve trap to remove water. UHP He was passed through a molecular sieve trap to remove water. ^{13}CO (Isotec Inc, 99% ^{13}C) was used as supplied.

Tylan mass flow controllers were used to regulate the flow of all four gases. The ^{12}CO and ^{13}CO streams flow from the flow controllers into two separate low dead-volume, 2-position, 4-port valves. One stream from each of these valves flows through a

bubblemeter to a vent line, while the other stream flows to a 4-way tee. He and D_2 are mixed and introduced into the third inlet of the 4-way tee, and the outlet from the tee goes to the microreactor. This flow scheme enables establishment of steady-state flow to the vent line in both CO streams and allows switching the reactor feed from one isotopically labeled form of CO to the other without significant flow perturbation. By simultaneous switching of the two 4-port valves, one avoids the elution of a small slug of the first isotope of CO held up in the valve and flow line.

2.3 Product Analysis

The steady-state distribution of reaction products was determined by gas chromatography and the temporal variation in the distribution of ^{12}C and ^{13}C in each product was determined by isotopic ratio gas chromatography-mass spectroscopy (28-30). Figure 1 shows a schematic of the analytical system. A 10-port GC sampling valve (Valco E410UWP) and a multiposition 16 sample-loop valve (Valco E6ST16T) were used to acquire and store samples. Both valves are housed in valve ovens and can be actuated electrically. Flow lines downstream of the reactor, the valve ovens and lines to the GC inlet were all maintained at 393 K, to minimize product condensation.

Product analysis is initiated by injecting the contents of one of the sample loops into a Perkin-Elmer Sigma 3B chromatograph containing a fused silica capillary column (0.25mm i.d. x 50 m)

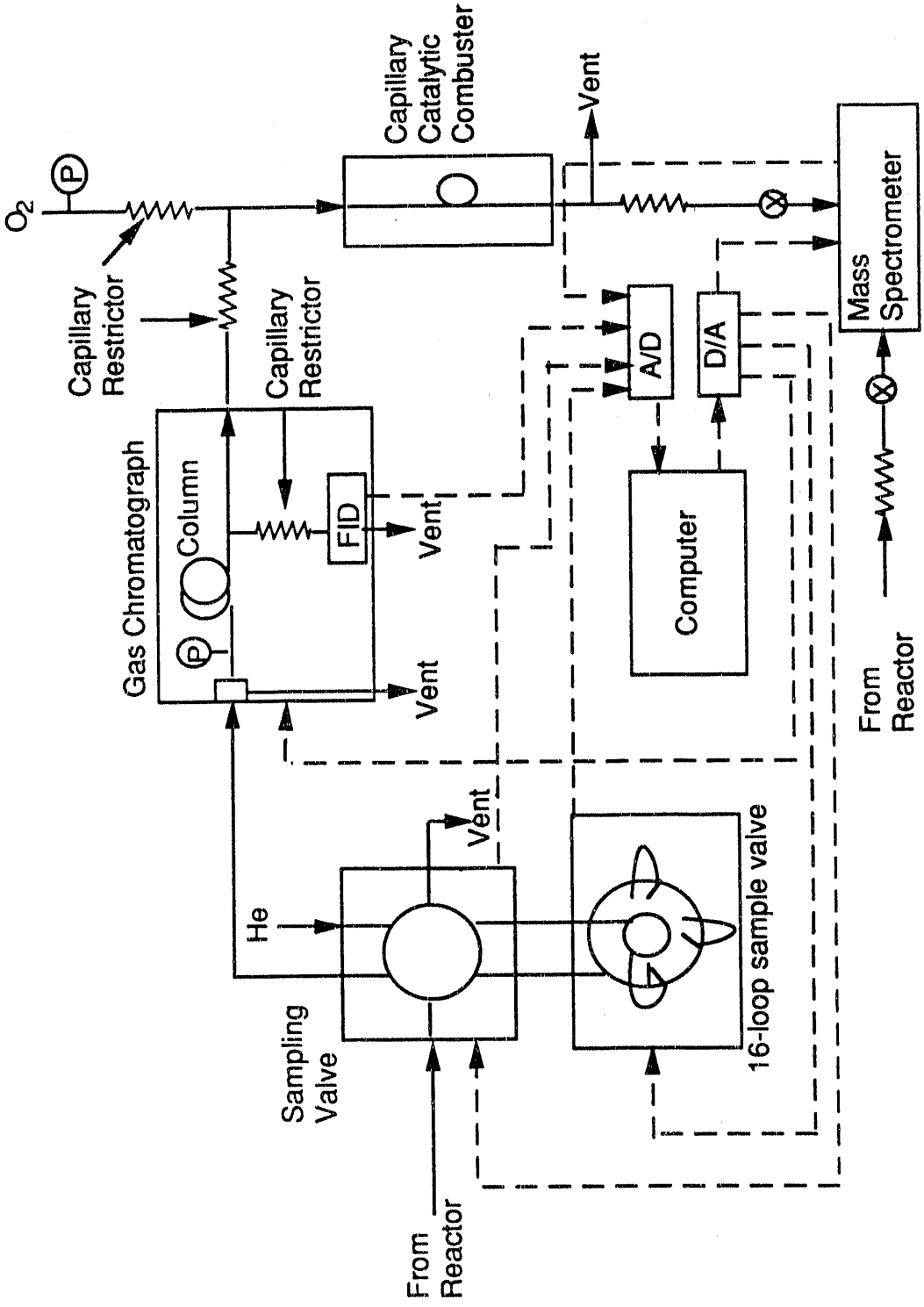

Fig. 1 Schematic of Analytical System

coated with a 1 μm film of SE-54. To achieve good product resolution, the column is maintained at 233 K for 4 min, then ramped at 20 K/min to 523 K, and finally held at 523 K for 10 min. This sequence results in a total sample analysis time of 28.5 min. C_{1-2} products elute at the initial temperature, C_{3-8} products elute during the temperature ramp, and C_{9-18} products elute at the final temperature. Products heavier than C_{14} hydrocarbons are not detected. The column effluent is split into two streams using a glass-lined capillary union (SGE); one line (0.11 mm i.d.) is sent to an FID detector for quantification, while the other transfer line (0.20 mm I.D.) is routed to a capillary combuster. This latter stream is mixed with O_2 and combusted to CO_2 in a 2 m long capillary (0.2 mm i.d.) containing Pt wire (0.13 mm o.d.) maintained at 873 K. Details of the capillary combuster are given in ref. (30).

The effluent from the combuster is leaked into a vacuum chamber containing the probe of a UTI 100C quadrupole mass spectrometer. The leak is accomplished by coupling the exit of the combuster directly to the inlet capillary restrictor of the mass spectrometer. The flow rate through the combuster is sufficient to preclude the back diffusion of air into the mass spectrometer. Complete combustion of all hydrocarbon products except methane was confirmed by the absence of peaks at masses corresponding to fragments derived from the hydrocarbons. The combustion of methane was found to be inhibited by the presence of a large amount of CO, which elutes at the same time as methane. The masses monitored by the mass spectrometer were amu 4 (He), amu 20 (D_2O

and $^{12}CD_4$), amu 21 ($^{13}CD_4$), amu 44 ($^{12}CO_2$) and amu 45 ($^{13}CO_2$). The distribution of ^{12}C and ^{13}C in methane was determined from the relative intensities of the signals for amu 20 and amu 21. The presence of D_2O in the products, which also produces a signal at amu 20, did not interfere with the detection of $^{12}CD_4$, since CD_4 and D_2O elute at sufficiently different times. The distribution of ^{12}C and ^{13}C in C_{2+} hydrocarbons was obtained from the relative signal intensities at amu 44 and amu 45. The temporal resolution of these products recorded by the mass spectrometer was comparable to that recorded by the FID.

Operation of the sampling valves was controlled by an IBM PC/XT computer programmed to acquire samples at preset times. The output from the FID detector and mass spectrometer were acquired by the computer at the rate of 2 data points a second.

2.4 Procedure

Experiments were carried out at temperatures between 453 K and 483 K. The ratio of D_2/CO was varied between 2 and 5, by holding the CO partial pressure at 0.1 atm, and varying the D_2 partial pressure between 0.2 and 0.5 atm. The total flowrate to the reactor was 100 cm^3/min. CO conversions were maintained as low as possible and, in most cases, were below 10%. For each set of conditions, the reaction was allowed to proceed in a mixture of ^{12}CO, D_2 and He for 20 min, to avoid the rapid initial loss in activity after start-up (27). Deactivation after the initial start-up proceeds with a first order decay time constant of 5 h. After each

experiment, the catalyst was reduced in D_2/He at 523 K for 2 h to ensure that steady-state activity levels were maintained.

Isotopic-tracer, transient response experiments were initiated by switching the feed from a stream containing ^{12}CO to one containing an equivalent concentration of ^{13}CO, after steady-state conditions had first been achieved in ^{12}CO. In this fashion, the steady-state activity of the catalyst was not perturbed by the switch in isotopic composition of the CO. To obtain good temporal resolution of the isotopic composition of the products, the following sampling sequence was used. A sample was taken 15 s after the switch in CO isotopic composition, followed by four samples taken at 10 s intervals, followed by a sample taken after an interval of 15 s, one taken after an interval of 20 s, one taken after an interval of 30 s interval, and two taken after intervals of 240 s. In this way, 11 samples are acquired in 10 min. Because the fill-time of the sample loops is 6s, data were not collected at intervals shorter than 10s. Since most changes in product isotopic composition occur in the first 2 min after the switch in feed isotopic composition, it is desirable to acquire data at intervals shorter than 10 s. To do so, the feed composition was changed from $^{13}CO/D_2$/He to $^{12}CO/D_2$/He and 5 samples were acquired at 10 s intervals for upto 60 s.

Figure 2 is a schematic of the data acquired during a typical experiment. The curves labelled $F(^{12}C)$ and $F(^{13}C)$ represent the fractions of the carbon atoms that are ^{12}C-labelled and ^{13}C-labelled, respectively. $F(^{12}C)$ is calculated by dividing the ^{12}C concentration observed at a given time by the initial steady-state ^{12}C

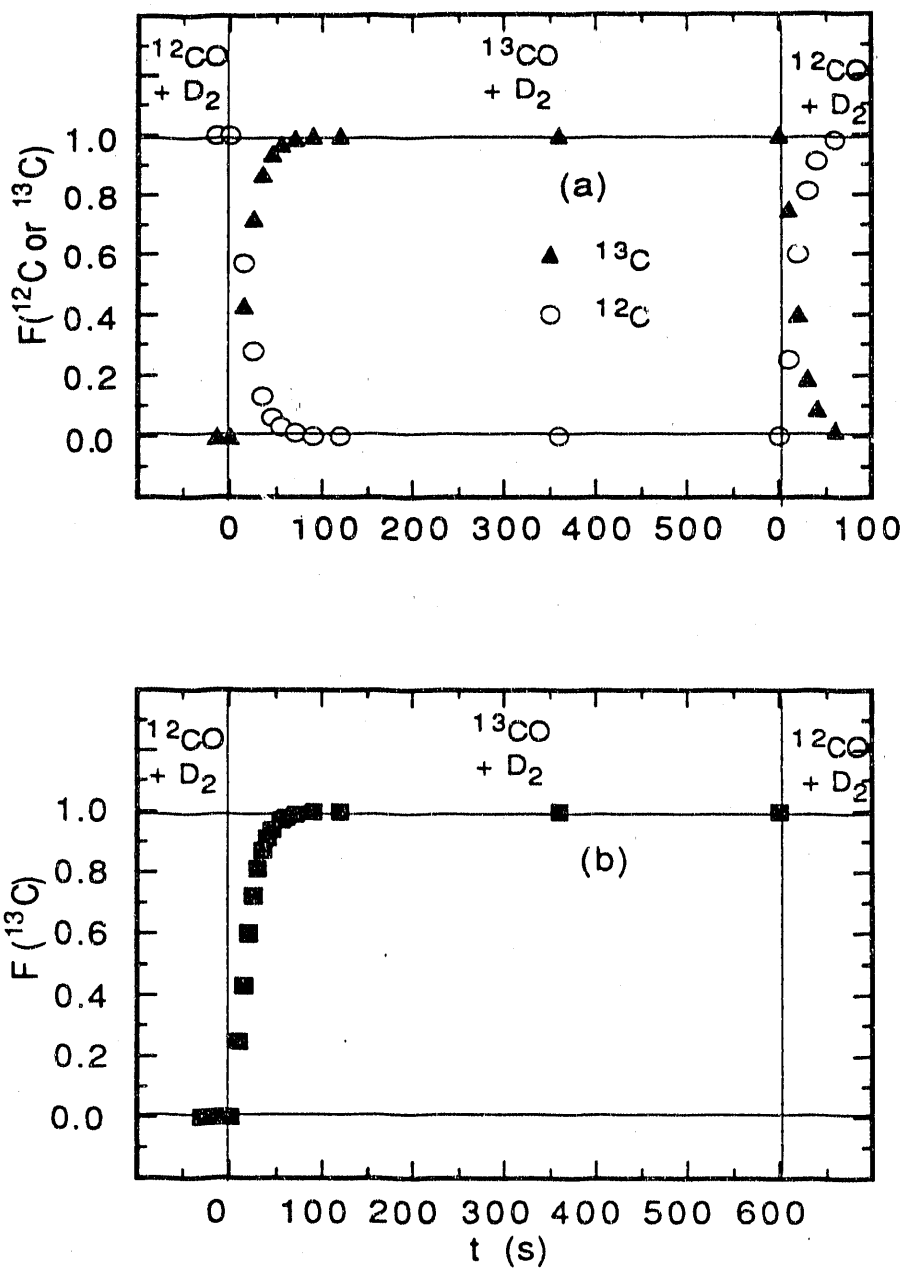

Fig. 2 (a) Isotopic Transient Data Sampling.

(b) Average Rise Curve.

concentration measured in $^{12}CO/D_2$; F(^{13}C) is obtained by dividing the observed ^{13}C concentration by the ^{13}C concentration measured after 10 min in $^{13}CO/D_2$. [It had previously been determined that the transients are complete in 10 min.] The sum of F(^{12}C) and F(^{13}C) for a given product always equals 1.0. For the first portion of the experiment, the rise curve points for each product were calculated as the average of the values of F(^{13}C) and [1-F(^{12}C)], and in the latter section, as the average of the values of F(^{12}C) and [1-F(^{13}C)]. All the points could then be combined to give a rise curve with 16 points. Sampling times were chosen such that data could effectively be collected at 5 s intervals for the first minute after the switch. Data acquired during the initial part of the experiment, as the ^{13}C content in the product rises, was found to be in agreement with data from the latter part of the transient experiment, when the ^{13}C content in the products declines.

3.0 THEORETICAL MODELLING

3.1 Chain growth model

Simulation of the experimentally observed transients was carried out on the basis of the scheme shown in Fig. 3. This scheme is identical to that used by Zhang and Biloen (14) and by Mims and McCandlish (16). In Fig. 3, CO_g and CO_s refer to gas-phase and adsorbed CO, $C_{m,s}$ refers to adsorbed monomeric building units and $C_{n,s}$ and $C_{n,g}$ refer to adsorbed alkyl chains and gaseous products containing n carbon atoms. Reference to reactions 1-16 in the

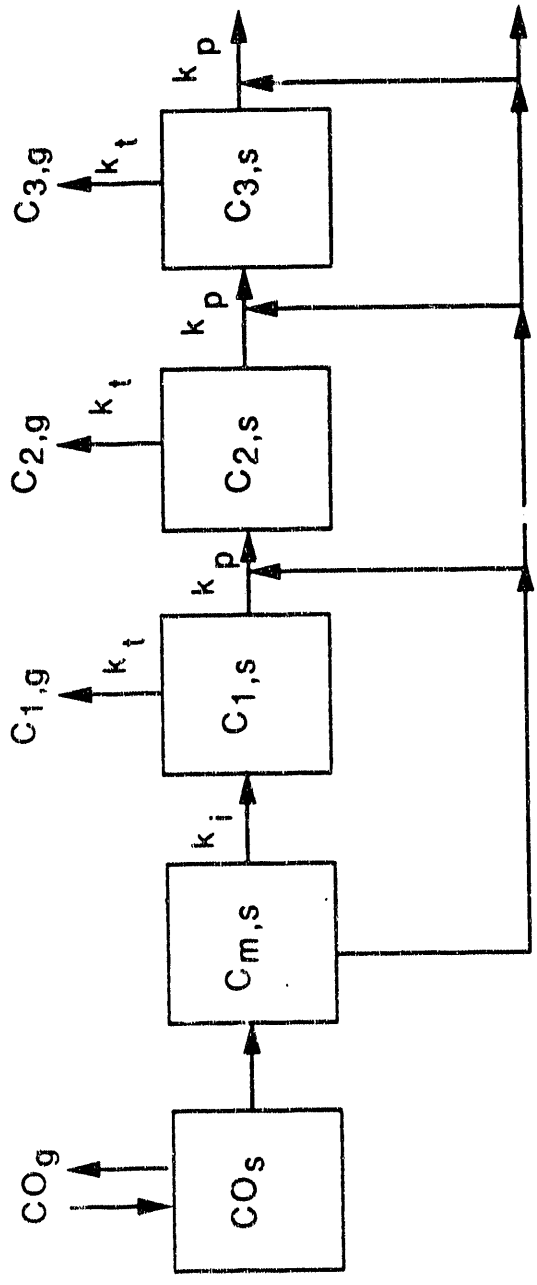

Fig. 3 Chain Growth Model

Introduction show that the scheme in Fig. 3 represents a simplification of the more detailed reaction sequence. First, the species C_s, CH_s and $CH_{2,s}$ are treated as a common species, C_m, on the assumption that the individual species are in equilibrium with each other. Second, no distinction is made between chain termination to olefins and paraffins. The rate coefficient for conversion of $C_{m,s}$ to $C_{1,s}$ is the apparent first-order rate coefficient for chain initiation, k_i. The rate coefficients k_p and k_t, are first-order rate coefficients for chain propagation, and termination. The dependence of any process on adsorbed hydrogen is not shown explicitly, since, for a given set of reaction conditions, the surface coverage of hydrogen is time independent.

3.2 Steady-State Rate and Label Balances

The scheme presented in Fig. 3 can be used to derive expressions for F_m and F_n, the fraction of labelled carbon in the monomer pool and in the pool of chains of length n, respectively. Equations for F_m and F_n are obtained in the following manner. A balance on labelled carbon entering and leaving the monomer pool gives

$$\theta_m \frac{dF_m}{dt} = k_d \theta_{CO} F_{CO} - \theta_m F_m \left(k_i + k_p \sum_{n=1}^{\infty} \theta_n \right) \qquad (1)$$

[In eqn. 1, θ_{CO}, θ_m, and θ_n are the surface coverages for CO, monomeric building units, $C_{m,s}$, and alkyl chains, $C_{n,s}$, respectively.] At steady state,

$$N_{CO} = k_d\theta_{CO} = k_i\theta_m + k_p\,\theta_m \sum_{n=1}^{\infty} \theta_n \qquad (2)$$

Combining eqns. (1) and (2), one obtains

$$\frac{d\,F_m}{d\,t} = \frac{F_{CO} - F_m}{\tau_m} \qquad (3)$$

where $\tau_m = 1/(k_i + k_p(\Sigma\theta_n))$

The appearance and disappearance of labelled carbon in the methane precursor pool C_1 is governed by

$$\theta_1\frac{dF_1}{dt} = k_i\theta_m F_m - k_t\,\theta_1 F_1 - k_p\,\theta_m\theta_1 F_1 \qquad (4)$$

Since at steady state

$$k_i\theta_m = k_t\,\theta_1 + k_p\,\theta_m\theta_1 \qquad (5)$$

eqn. (4) can be rewritten as

$$\frac{dF_1}{dt} = \frac{F_m - F_1}{\tau} \qquad (6)$$

where $\tau = 1/(k_p\theta_m + k_t)$. The appropriate balance on labelled carbon in the pool of alkyl species containing n carbon atoms is given by

$$n\theta_n\frac{dF_n}{dt} = k_p\theta_m\theta_{n-1}[(n-1)F_{n-1} + F_m] - k_p\theta_m\theta_n(nF_n) - k_t\theta_n(nF_n) \qquad (7)$$

Once again, at steady state,

$$k_p\theta_m\theta_{n-1} = k_p\theta_m\theta_n + k_t\theta_n \qquad (8)$$

so that eqn. 7 can be rewritten as

$$\frac{d\,F_n}{d\,t} = \frac{\frac{(n-1)F_{n-1} + F_m}{n} - F_n}{\tau} \qquad (9)$$

The initial conditions for F_m, F_1, and F_n are $F_m = F_1 = F_n = 0$ at $t = 0$. Since CO is assumed to equilibrate rapidly with the catalyst

surface, $F_{CO} = 1$ for $t \geq 0$. With these initial conditions, the solution to eqn. 6 is

$$F_m(t) = 1-\exp(-t/\tau_m) \qquad (10)$$

An analytical solution for $F_n(t)$ can be obtained using Laplace transforms and the convolution theorem (31). The resulting expression is

$$F_n(t) = 1.0 + \left(\frac{\exp(-\frac{t}{\tau_m})}{n}\right)\sum_{i=1}^{n}(-1)^{i-1}\left(\frac{\tau_m}{\tau-\tau_m}\right)^{i}$$
$$+ \left(\frac{\exp(-\frac{t}{\tau})}{n}\right)\sum_{i=1}^{n}\frac{1}{\tau^i}\sum_{r=0}^{i-1}\frac{t^{i-r-1}}{(i-r-1)!}\tau^{r+1}\left(\left(-\frac{\tau_m}{\tau-\tau_m}\right)^{r+1}-1\right) \qquad (11)$$

4.0 RESULTS AND DISCUSSION

4.1 Catalyst Activity and Selectivity

Gas chromatographic analyses of the products taken during the isotopic transient experiments shows that the catalyst activity and selectivity are unchanged over the 11-min duration of the experiment. The main products observed are α-olefins, cis- and trans-β-olefins and n-paraffins. A few branched products are also detected, but these constitute less than 10% of the total product. Alcohols and other oxygenates are not detected. Mass spectometric analysis of the products indicates that no CO_2 is formed and that D_2O is the only oxygenated product.

Figure 4 shows an Anderson-Schulz-Flory (ASF) plot of the turnover frequencies for C_1-C_{12} at 463K and a $D_2/CO = 3$. As is characteristic of Ru catalysts (3, 30), the points for C_2 and C_3

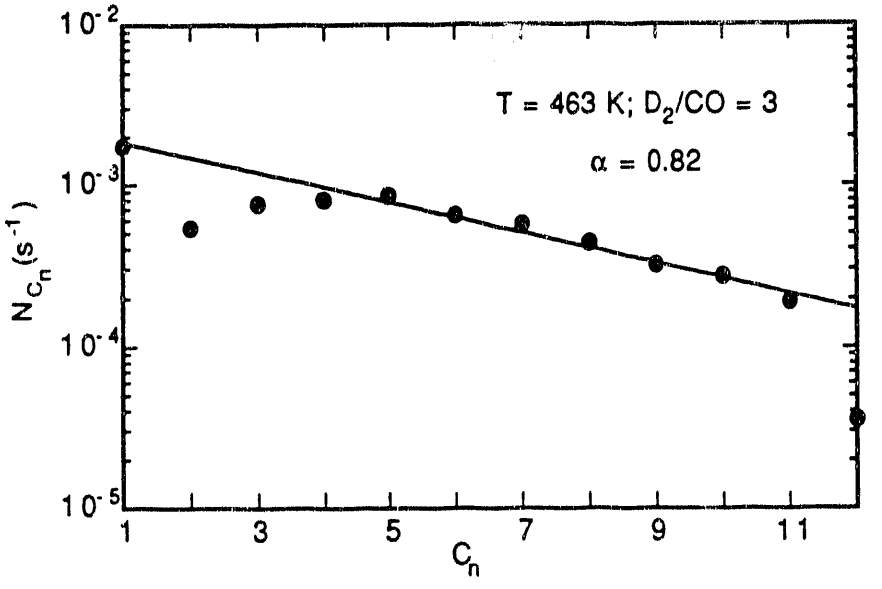

Fig. 4 Anderson-Schulz-Flory plot, log N_{Cn} vs Carbon number, where N_{Cn} is the turnover frequency for products of carbon number n.

products fall below the line passing through the points for the C_4-C_{11} products. From the slope of this line, α is determined to be 0.82.

Figures 5a and 5b show the dependence of the turnover frequency for CO consumption and α on temperature, respectively. N_{CO} is calculated as $\Sigma n N_{Cn}$ for values of n between 1 and 13, and α is taken as the slope of the linear portion of the ASF plot. It is seen from Fig. 5a that N_{CO} exhibits an Arrhenius behavior for temperatures below 473 K, and then becomes nearly constant at higher temperatures. Calculations of the Weisz parameter indicate the absence of intraparticle mass transfer limitations, leading to the conclusion that the observed dependence on temperature is a reflection of the intrinsic kinetics. Figure 5b shows that α decreases monotonically from 0.88 to 0.73 as temperature increases from 453K to 498K.

The effects of D_2 partial pressure on N_{CO} and α, for a fixed temperature and CO partial pressure are shown in Figs. 6a and 6b. The value of N_{CO} is seen to increase linearly with increasing D_2 partial pressure, whereas the value of α decreases monotonically. The trends reported in Figs. 5b and 6b are similar to those observed previously for Ru/Al_2O_3 (3) and for supported Fe catalysts (4, 5).

Figure 7 shows a series of product selectivity plots. It is observed that the total straight-chain olefin/n-paraffin ratio rises from C_2 to C_4 and then decreases with increasing carbon number. The fraction of internal olefins in the straight-chain olefin product rises with increasing carbon number. The β-olefin fraction in the

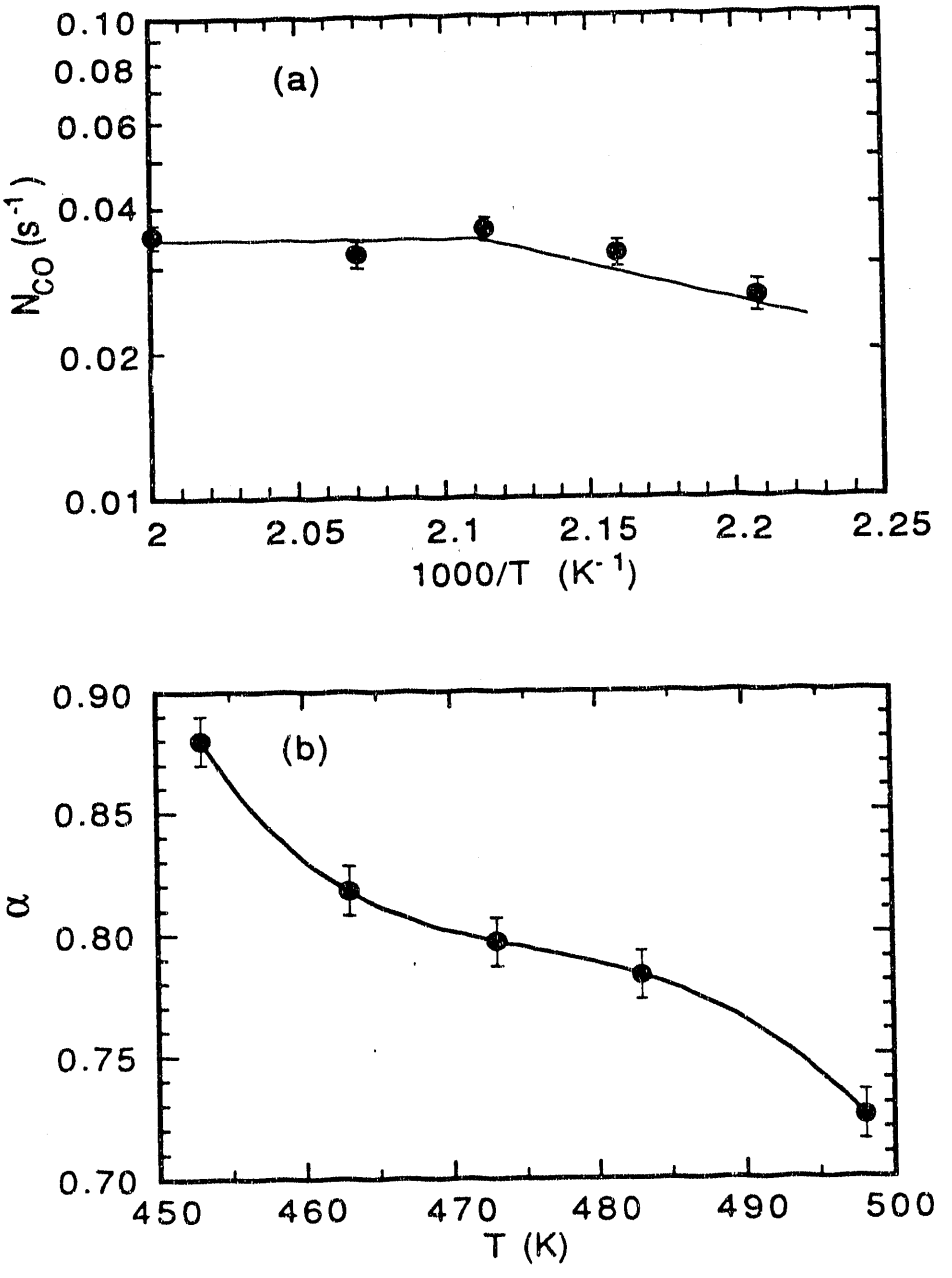

Fig. 5 a) CO turnover frequency, N_{CO}, as a function of temperature; (b) Chain growth parameter α as a function of temperature; $D_2/CO = 3$.

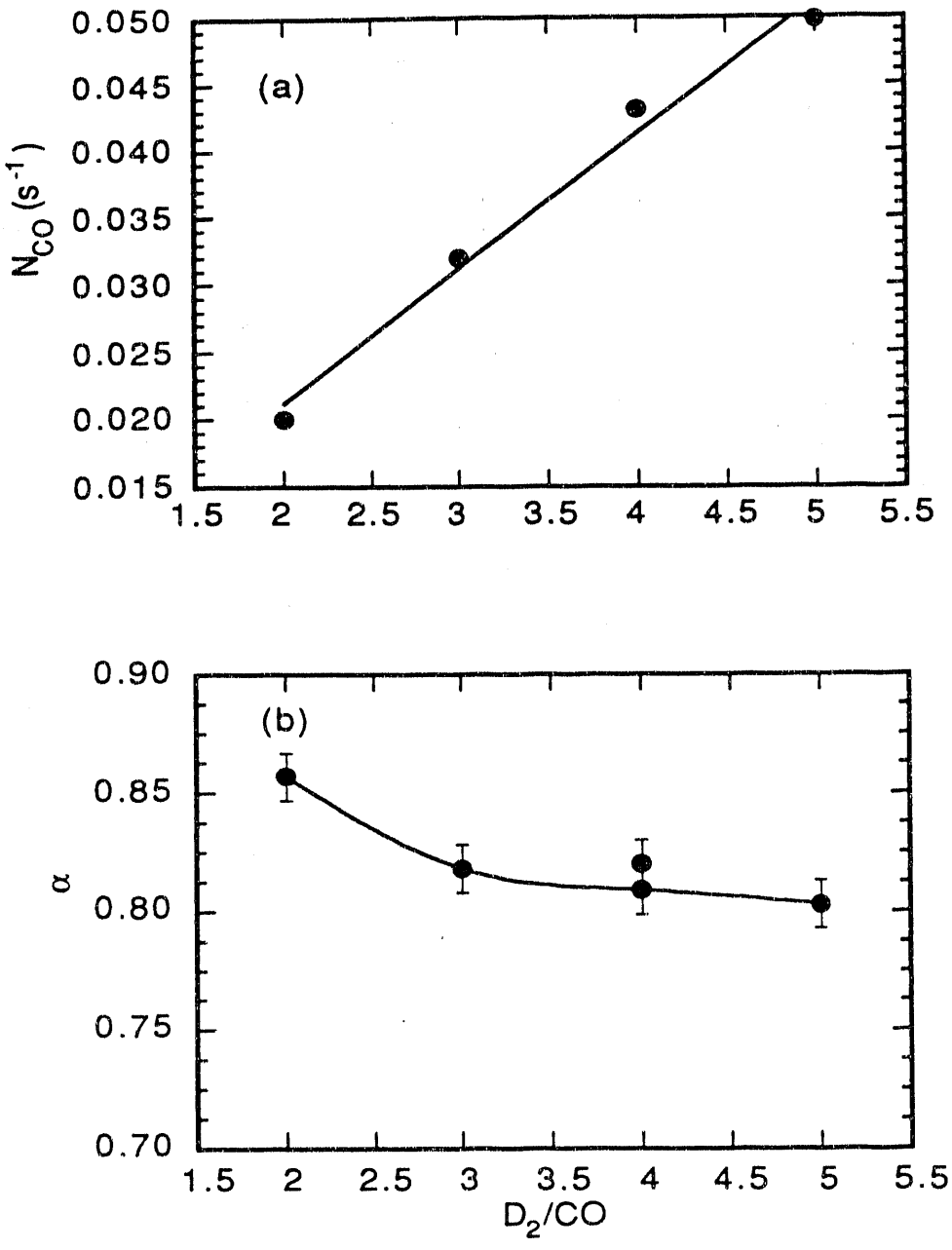

Fig. 6 (a) CO turnover frequency, N_{CO}, as a function of D_2/CO ratio; (b) Chain growth parameter α as a function of D_2/CO ratio; T = 463 K.

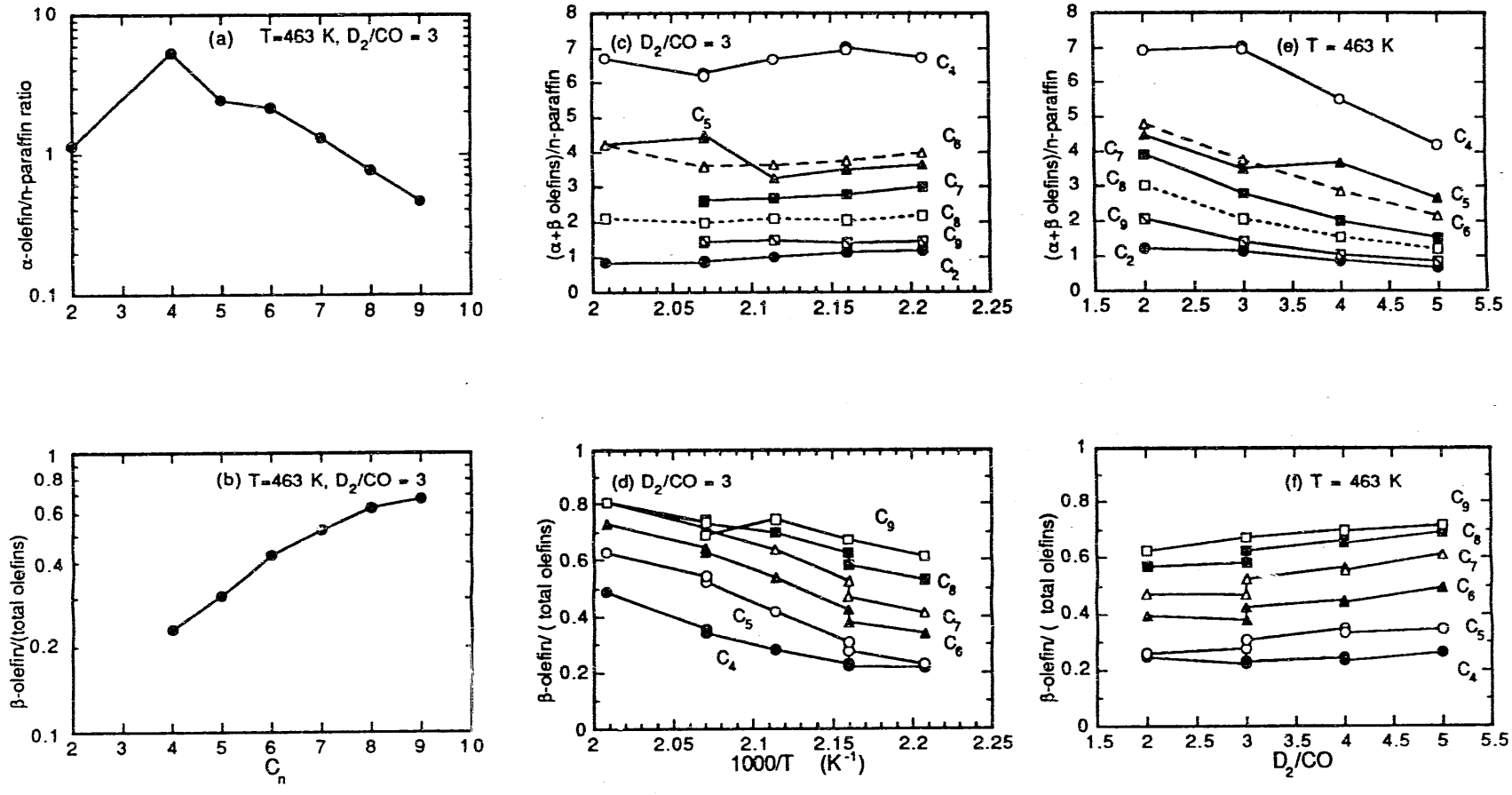

Fig.7 (a) α-olefin to n-paraffin ratio vs Carbon Number; (b) β-olefin fraction in straight-chain olefins vs Carbon Number; T = 463 K, D_2/CO = 3; (c) Straight chain olefins to n-paraffin ratio as a function of temperature; (d) β-olefin fraction in straight-chain olefins as a function of temperature; D_2/CO = 3; (e) Straight chain olefins to n-paraffin ratio as a function of D_2/CO ratio; (f) β-olefin fraction in straight-chain olefins as a function of D_2/CO ratio; T= 463 K.

olefin product at any carbon number also increases with temperature. The ratio of olefins to paraffins decreases with increasing D_2/CO ratio. These changes in product selectivity are in agreement with data reported by Dictor and Bell (32) on an Fe catalyst and by Schulz (33) on a Fe/Mn catalyst.

4.2 Isotopic Transients

The incorporation of ^{13}C and the concurrent decline in ^{12}C in the C_1 and C_{3-8} products was monitored by isotope-ratio GC-MS, following a switch in the feed from $^{12}CO/D_2$ to $^{13}CO/D_2$. The concentration of the C_2 product was too low for mass spectrometric detection. Figures 8 and 9 show representative plots of $F_1(t)$ and $F_n(t)$ (n = 3-8), respectively. The data points in Fig. 8 are based on the measured isotopic fractions in all olefins and paraffins of a given carbon number, since it was observed that the dynamics for products with a given number of carbon atoms could not be differentiated on the basis of structure (i.e., olefin vs paraffin, α-olefin vs β-olefin, n-paraffin vs branched paraffin). Figure 8 shows that the methane transient rises rapidly but on comparison with Figure 9, does not lie before the C_3 data. The transients for C_{3+} are shown in Fig. 8 and are seen to approach 1.0 after 120 s. While there is considerable scatter in the data, it is observed that the appearance of ^{13}C-labelled carbon in the products is progressively slower as the value of n increases from 3 to 8. This trend is particularly evident for the C_{6+} products.

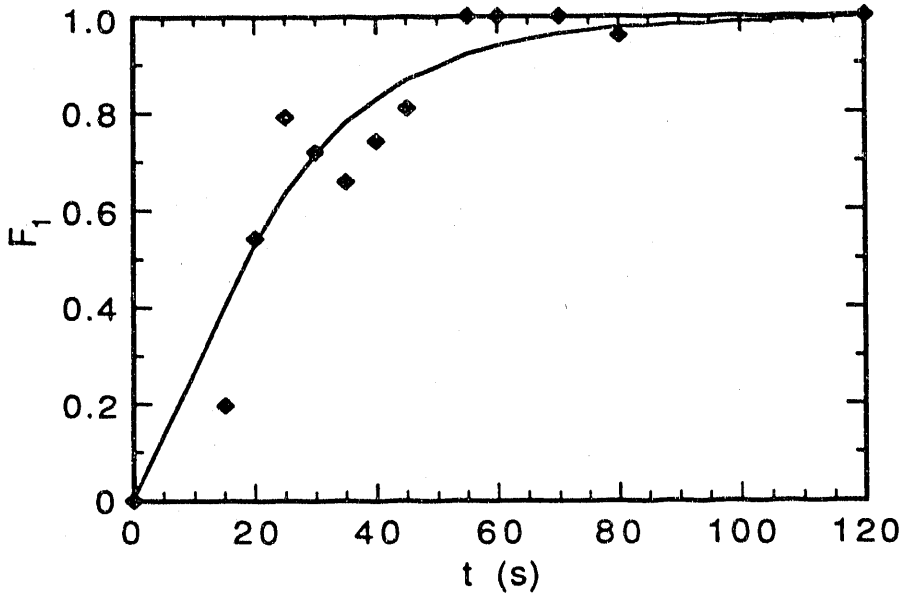

Fig. 8 F ($^{13}CD_4$) rise after a switch from $^{12}CO/D_2$ to $^{13}CO/D_2$ at t = 0. T = 463 K, D_2/CO = 3.

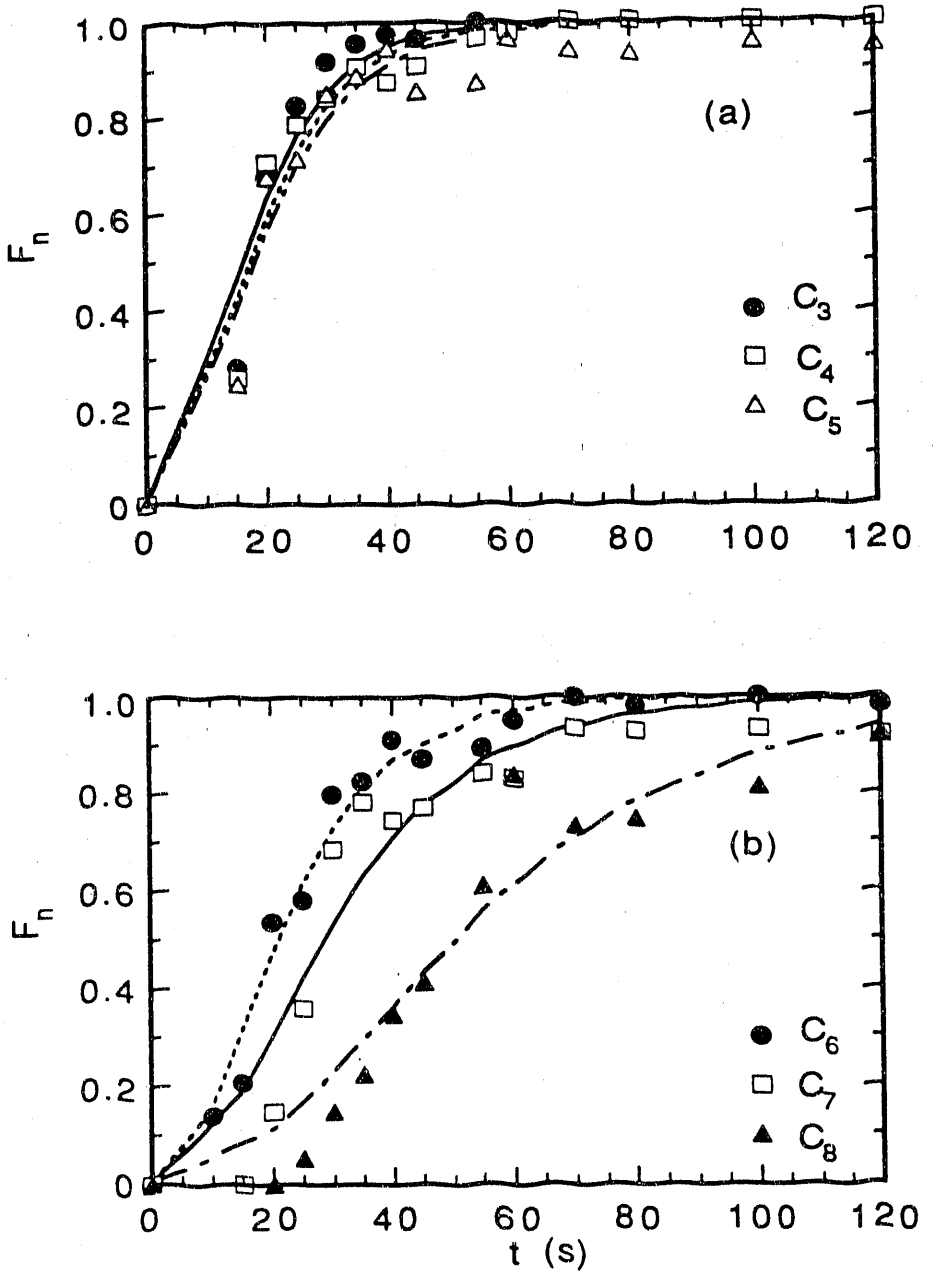

Fig. 9 (a) $F(^{13}C)$ rise in the C_3-C_5 products; (b) $F(^{13}C)$ rise in the C_6-C_8 products; transient response after a switch from $^{12}CO/D_2$ to $^{13}CO/D_2$ at t = 0. T = 463 K, D_2/CO = 3.

4.21 Parameter evaluations

Values of the two parameters τ and τ_m were obtained by fitting the analytical expressions for F_n given by eqn. 11 to the experimental data. This was done using a quasi-Newton method (31) to minimize the objective function $S_n(\tau, \tau_m)$ defined as

$$S_n(\tau, \tau_m) = \sum_{j=1}^{M} [F_n^{expt}(t_j) - F_n(t_j)]^2 \qquad (12)$$

In eqn. 12, $F_n^{expt}(t_j)$ and $F_n(t_j)$ are the experimental and theoretical values at time t_j and M is the number of experimental points.

Figures 8 and 9 show data and model fits at a temperature of 463 K and a D_2/CO ratio of 3. The F-test was used to assess the statistical adequacy or lack of fit of the model (34, 35). For each value of n, the F-test indicated that the model fit the data at a 95% confidence interval. The parameters τ and τ_m were determined independently for each of the transients from C_1-C_8. Table 1 shows values obtained for the data set at 463 K and D_2/CO = 3. It is evident that the values of τ and τ_m determined from different transients exhibit a modest spread in values. Of particular note are the large values for τ_m for n = 1 and n = 8. For n = 8, the value of τ_m is directly related to the slow approach of $F_8(t)$ to unity (see Fig. 8), as a consequence of a partial loss of product due to condensation in the transfer-line upstream of the capillary combuster. The large value of τ_m for n =1 also reflects the slow approach of $F_1(t)$ to unity. In part this may be due to the presence of ^{12}C impurity in the ^{13}CO

TABLE 1

τ and τ_m obtained from fitting the chain growth model to the data: $D_2/CO = 3$; $T = 463$ K.

Carbon Number	τ(s)	τ_m(s)
1	4.7	19.6
3	5.2	7.8
4	3.7	10.3
5	2.8	12.3
6	3.2	12.7
7	3.5	16.7
8	5.7	31.6
Average	3.9 ± 0.9[1]	11.9 ± 4.0[2]

1 Average of values for C_1, C_{3-7}. See text for details.

2 Average of values for C_{3-7}.

feed. For these reasons, it was decided that average values of τ should be based on the individual values obtained for C_1 and C_{3+}, and average values of τ_m should be based on individual values obtained for C_{3+}. It was found that the F_n predictions based on the average values of τ and τ_m were nearly indistiguishable from those based on the best-fit model values for each n from 3-7. Tables 2 and 3 show the average values of τ and τ_m obtained for the transients when temperature and the D_2/CO ratio were varied.

From Tables 2 and 3, it is seen that the modelling leads to the result that $\tau < \tau_m$ and that the ratio τ_m/τ lies between 3 and 13. This is similar to the findings of Mims *et al.* (16) on Co/SiO_2 and promoted Fe and is in disagreement with the results of Zhang and Biloen (14) on Co. This modelling result implies that active species on the surface spend a larger portion of their residence time in the monomer pool than in alkyl chains and further translates to higher coverages by the monomer pool than by the hydrocarbon precursors.

4.22 Intrinsic Rate Parameters and Surface Coverages

The values of k_i, k_p, and k_t, and θ_m and $\Sigma\theta_n$, can be determined from the values of τ and τ_m, obtained through the fitting procedure described above, and the values of N_{CO}, N_{Cn} and α, obtained from steady-state rate data. The approach used to calculate the rate coefficients and the surface coverages is described in the Appendix. Table 4 gives a complete summary of all the results at 463 K and $D_2/CO = 3$.

TABLE 2

Average τ and τ_m obtained from fitting the chain growth model to the data:
D_2/CO = 3; temperature is varied.

T (K)	τ_{av}(s)	$\tau_{m,av}$(s)
453	3.5	13.5
463	3.9	11.9
463	3.9	15.0
473	1.8	18.1
483	2.1	18.8
483	1.6	19.1

TABLE 3

Average τ and τ_m obtained from fitting the chain growth model to the data:
D_2/CO ratio is varied; T = 463 K

D_2/CO	τ_{av}(s)	$\tau_{m,av}$(s)
2	3.3	14.0
3	3.9	11.9
4	1.5	16.7
4	2.5	15.2
5	2.5	13.1

TABLE 4

Transient at standard reaction conditions: Ru/TiO_2 catalyst.

Reaction T= 463 K; $D_2/CO = 3$

$N_{CO} = 0.032\ s^{-1}$; $\alpha = 0.82$

$\tau = 3.9$ s

$k_p^{app} = 0.212\ s^{-1}$, $0.208\ s^{-1}$

$k_t = 0.047\ s^{-1}$, $0.046\ s^{-1}$

$k_i = 0.022\ s^{-1}$, $0.021\ s^{-1}$

$\Sigma\theta_n = 0.14$ ML, 0.17 ML

$\theta_m = 0.38$ ML, 0.48 ML

$\theta_{CO} = 0.67$ ML

(Two values indicate results of separate experiments.)

Figure 10 shows the variation of rate coefficients k_i, k_p, and k_t with T^{-1}. The value of the initiation rate coefficient k_i appears to be independent of temperature. This trend can be rationalized if the initiation of chain growth is assumed to proceed via the process $CH_{2,s} + H_s \rightarrow CH_{3,s}$. In such a case, k_i is an apparent rate coefficient, representing the product of the true rate coeffcient, k'_i, and the coverage of adsorbed hydrogen, θ_H. Since k'_i is expected to increase with increasing temperature but θ_H is expected to decrease, the product of k'_i and θ_H should be less sensitive to temperature than either of the two factors making up the product. The plots of k_p and k_t versus T^{-1} both have negative slopes, from which it is determined that the activation energy for chain propagation, E_p, is 8 kcal/mol and the activation energy for chain termination, E_t, is 20 kcal/mol.

The present evaluation of k_p can be compared to the estimated apparent rate of propagation reported by Zhang and Biloen (14). To do so, k_p is multipled by θ_m (see below) to obtain $k_p^{app} = k_p\theta_m$. For the data presented in Fig. 10, k_p^{app} lies between 0.25 and 0.5 s^{-1}.This range lies somewhat below Zhang and Biloen's estimate of >1 s^{-1}. There is close agreement between the values of k_t reported here and in the work of Yokomizo *et al.* (21). Using a Ru/TiO_2 catalyst similar to that described here, Yokomizo *et al.* determined a value of k_t = $0.044s^{-1}$ at T = 463 K, from modelling of temperature programmed surface reaction (TPSR) data. For the same reaction conditions, we obtain a value of k_t = 0.047 s^{-1}.

The activation energy for chain termination (20 kcal/mol) is

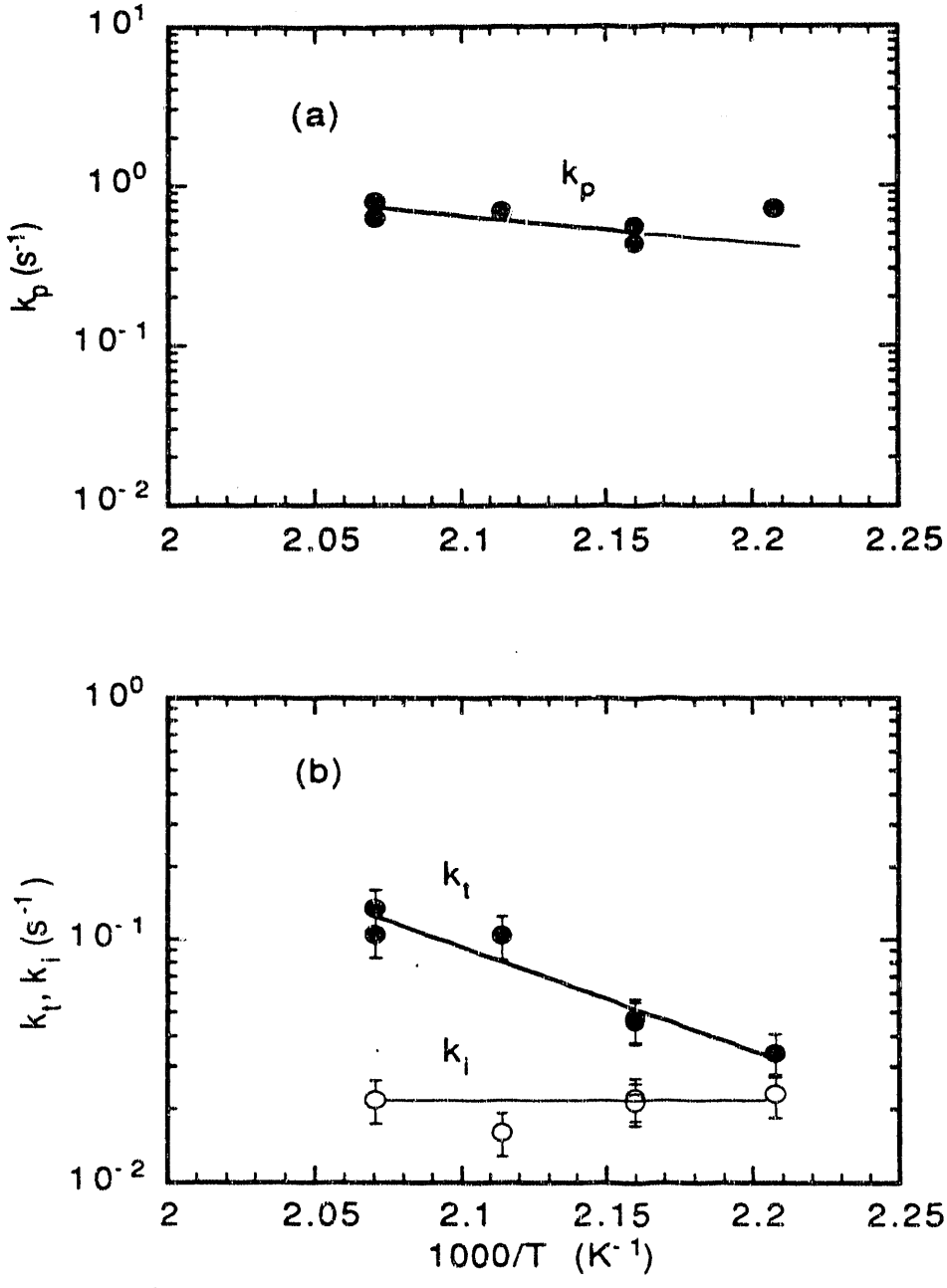

Fig. 10 (a) Propagation rate constant, k_p, as a function of temperature; (b) Inititation rate constant, k_i, and termination rate constant, k_t, as a function of temperature; $D_2/CO = 3$.

significantly higher than that for chain propagation (8 kcal/mol). This explains why the value of α decreases with increasing temperature (see Fig. 5b). While experimental values of E_t and E_p have not been reported in the literature previously, Shustorovich and Bell (36) have shown using the Bond-Order-Conservation-Morse-Potential (BOC-MP) approach, that E_t should be larger than E_p for Ni and Fe. Thus, for Ni(111), E_p for the chain growth step to form a C_2 species is estimated to be 6 kcal/mol, whereas E_t is estimated to be 11 kcal/mol for chain termination to ethylene and 19 kcal/mol for chain termination to ethane. For Fe(110), E_p for the formation of C_2 species is estimated to be 20 kcal/mol, whereas E_t is estimated to be 16 kcal/mol for chain termination to ethylene and 32 kcal/mol for chain termination to ethane.

The effects of temperature on the calculated values of surface coverage by carbonaceous species is given in Fig. 11 and Table 5. The values of θ_m lies between 0.3 and 0.6 ML, while the value of $\Sigma\theta_n$ lies between 0.12 and 0.05 ML. It is observed that with increasing temperature, θ_m increases, but θ_1 and $\Sigma\theta_n$ decrease. These trends are consistent with the observed temperature dependences for chain initiation, propagation, and termination. As seen in Fig. 10, increasing temperature has virtually no effect on k_i, but increases both k_p and k_t. Since E_t is larger than E_p, the rate of chain termination increases more rapidly than the rate of chain propagation, with the result that the surface coverage by growing alkyl chains decreases, whereas the coverage by monomeric building units increases. The total coverage by reaction intermediates is

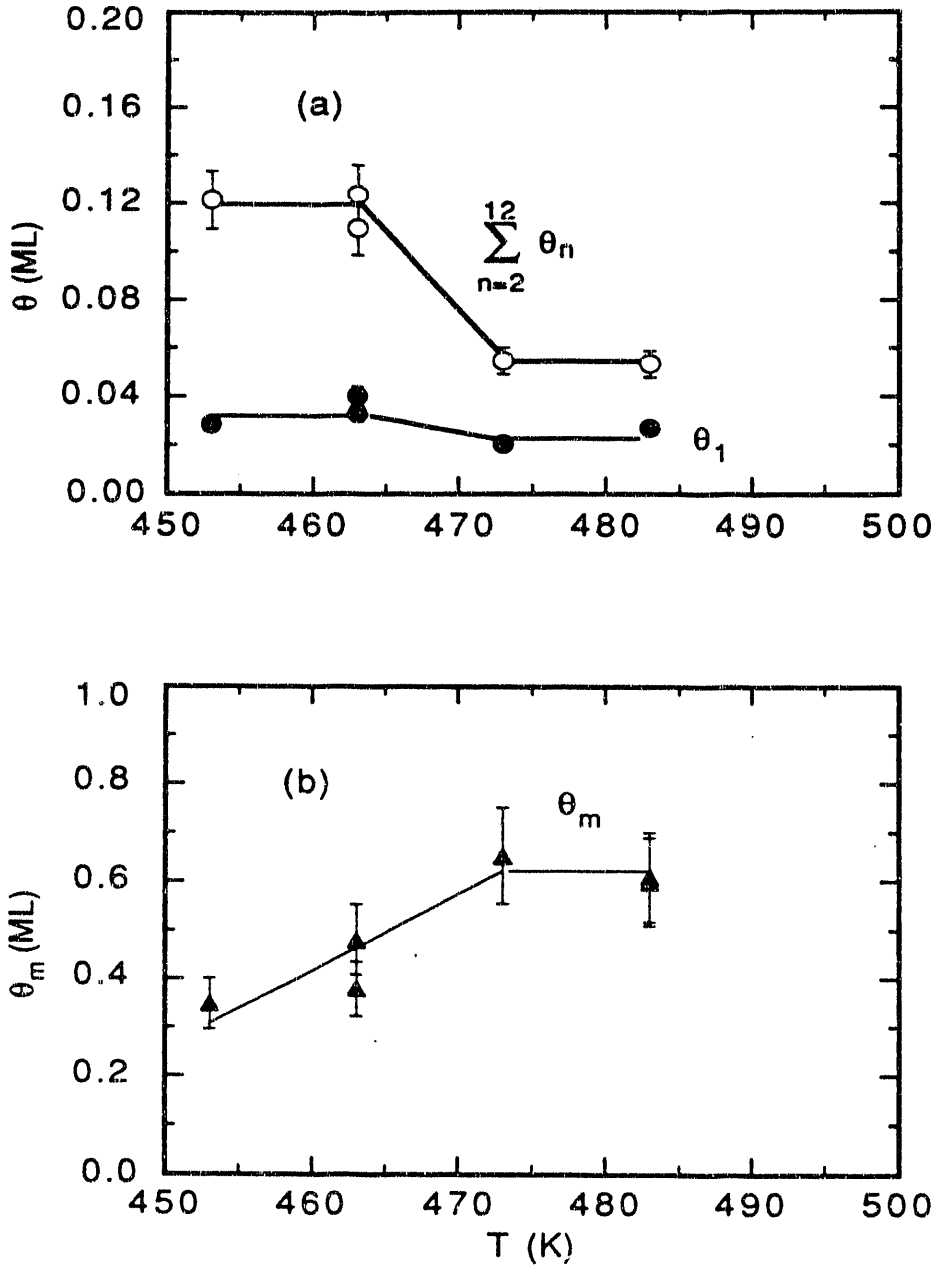

Fig. 11 (a) Coverage by alkyl surface species as a function of temperature; (b) Coverage by monomeric species as a function of temperature; $D_2/CO = 3$.

TABLE 5

Coverage by reaction intermediates:
D_2/CO = 3, reaction temperature is varied.

T (K)	θ_1(ML)	$\sum_{n=2}^{12} \theta_n$ (ML)	θ_m(ML)	Total $\theta_{(n+m)}$(ML)	θ_{CO}(ML)
453	0.03	0.12	0.35	0.5	-
463	0.03	0.11	0.38	0.52	0.67
463	0.04	0.12	0.48	0.64	0.67
473	0.02	0.05	0.62	0.69	-
483	0.03	0.05	0.61	0.69	0.65
483	0.03	0.61	0.60	0.68	0.65

seen to be between 0.5 and 0.7 of a monolayer.

To determine whether or not the surface coverages estimated from the model are representative of actual coverages by carbonaceous species, a series of temperature-programmed surface reactions were carried out. Using the techniques described in ref. (27), measurements were made of the CO uptake capacity of the freshly reduced catalyst, the CO uptake of the catalyst after 20 min under reaction conditions, and the total carbon inventrory on the catalyst surface, exclusive of CO, after 20 min under reaction conditions. The room-temperature uptake of CO on the freshly reduced catalyst is 1.3 ML. After 20 min of reaction at 463 K and a D_2/CO ratio of 3, the CO coverage decreases to 0.67 ML and the total amount of carbon exclusive of CO is 1.08 ML. The last of these figures compares very favorably with the value of $\Sigma n\theta_n = 1.03$ obtained from the simulation of the transient response experiments. Moreover, the sum of θ_{CO} plus $\Sigma\theta_n$ for n=1-13 is equal to 1.19-1.31 ML (see Table 5), in good agreement with the initial CO uptake capacity of the catalyst, 1.3 ML. A similar level of agreement is observed when the reaction is carried out at 483 K.

The close correspondence between the predicted and experimentally observed coverages by carbon indicates that the dynamics of the transient response experiments are governed primarily by the α (carbidic) and β' (growing alkyl chains) forms of surface carbon (21). As noted by Krishna and Bell (27), these forms of carbon exhibit time constants that are much shorter than that associated with the build up and consumption of β'' carbon, which

contribute to the slow deactivation of Ru.

The observation of a higher surface coverage by monomeric units than by growing chains is qualitatively consistent with the findings of Yokomizo *et al.* (21) for Ru/TiO_2 and Mims *et al.* (16) for Co/Al_2O_3. Yokomizo *et al.* (21) estimate the surface coverage by all C_1 species to be 0.25 ML, and the coverage by C_{2+} chains to be about 0.10 ML. This compares favorably with the present results which indicate a surface coverage of 0.12 ML by C_{2+} species and a coverage of 0.4 by all C_1 species. Since the estimates of surface coverages reported here and by Yokomizo *et al.* (21) are based on completely different methods, the degree of agreement is all the more remarkable.

Figure 12 indicates the effects of D_2 partial pressure on k_i, k_p, and k_t. The value of k_p is essentially independent of the partial pressure of D_2, consistent with what would be expected, since chain growth does not involve adsorbed D atoms. By contrast, the value of k_t rises linearly with increasing D_2 partial pressure. This trend can be explained in the following manner. As defined, k_t is an apparent rate coefficient representing termination to olefinic and paraffinic products. Expressing k_t in terms of its component parts gives

$$k_t = \bar{k_t}\theta_v + \dot{k_t}\theta_D \qquad (13)$$

where $\bar{k_t}$ and $\dot{k_t}$ are the rate coefficients for termination to olefins and paraffins, respectively, and θ_v and θ_D are the vacancy and D atoms surface coverages, respectively. The increase in k_t with D_2 partial pressure can, therefore, be attributed to the increase in θ_D.

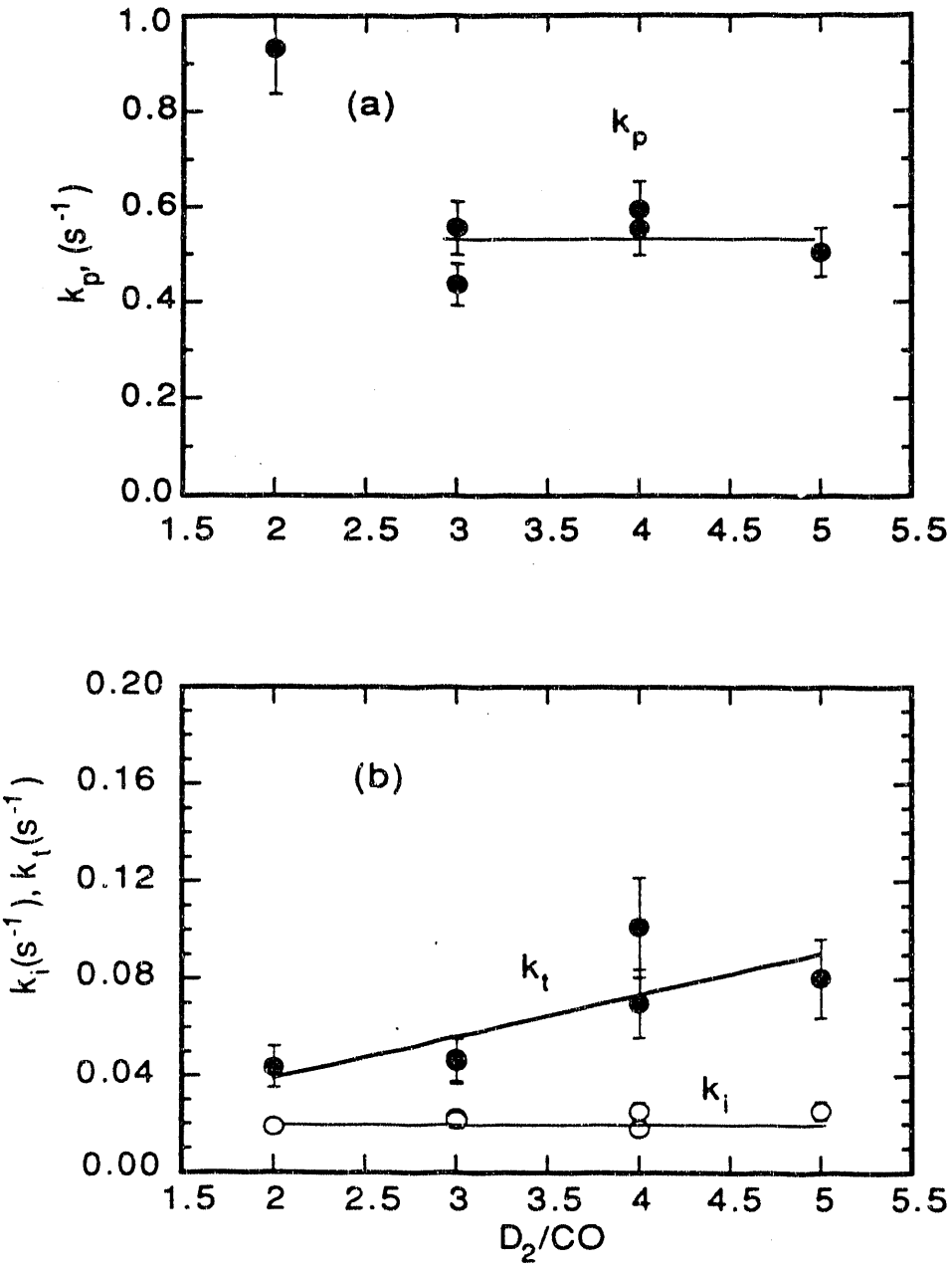

Fig. 12 (a) Propagation rate constant, k_p, as a function of D_2/CO ratio; (b) Inititation rate constant, k_i, and termination rate constant, k_t, as a function of D_2/CO ratio

In agreement with this, the product distribution shifts to more paraffinic products as the D_2 partial pressure rises (see Fig. 7e). Equation 13 also helps explain the weak dependence of the olefin to paraffin ratio on temperature, seen in Fig. 7c. Since θ_D decreases and k_t^- increases with temperature, the product of θ_D and k_t^- should show a weaker temperature dependence.

The independence of k_i on D_2 partial pressure is puzzling, since as discussed above, this rate coefficient is, in fact, the product of an intrinsic rate coefficient and θ_D. No explanation can be given for why k_i does not behave in the same manner as k_t.

The dependence of the surface coverages of carbonaceous species calculated from the model is given in Fig. 13 and Table 6. The coverage by monomeric building units and C_1 chain initiators increases with D_2 partial pressure and approaches a constant value, whereas the surface coverage by C_{2-12} chains appears to pass through a maximum. The trends in θ_m and θ_1 indicate that with decreasing D_2 partial pressure, the rate of conversion of nascent carbon atoms to CH_x species is somewhat faster than the consumption of these species to form reaction products. The total coverage by reactive species is seen to increase from 0.4 to 0.8 ML as the D_2/CO ratio is changed.

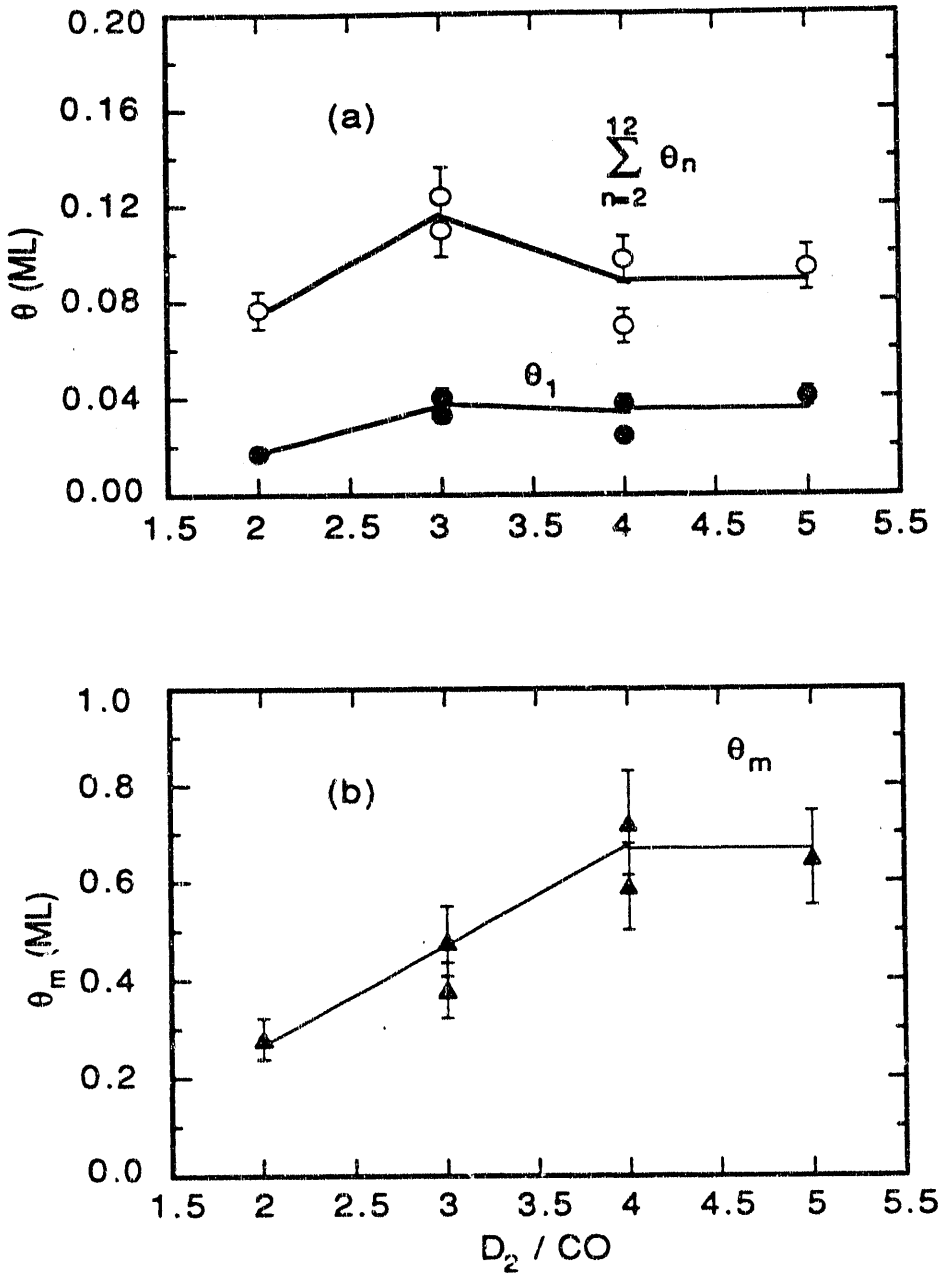

Fig. 13(a) Coverage by alkyl surface species as a function of D_2/CO ratio; (b) Coverage by monomeric species as a function of D_2/CO ratio; T = 463 K.

TABLE 6

Coverage by reaction intermediates:
D_2/CO ratio is varied, T= 463 K.

D_2/CO	θ_1(ML)	$\sum_{n=2}^{12} \theta_n$ (ML)	θ_m(ML)	Total $\theta_{(n+m)}$(ML)
2	0.03	0.12	0.28	0.43
3	0.03	0.11	0.38	0.52
3	0.04	0.12	0.48	0.64
4	0.02	0.07	0.72	0.81
4	0.04	0.10	0.59	0.73
5	0.04	0.10	0.65	0.79

CONCLUSIONS

Isotopic tracer methods have been used to determine the dynamics of chain initiation, propagation and termination for Fischer-Tropsch synthesis over Ru/TiO_2. Values for the rate coefficients k_i, k_p and k_t were determined by fitting theoretically generated transient response curves to those obtained experimentally. The rate coefficient for chain initiation, k_i, is independent of both temperature and D_2/CO ratio. The rate coefficient for chain propagation, k_p, has an activation energy of 8 kcal/mol, but is independent of the D_2/CO ratio. The rate coefficient for chain termination, k_t, has an activation energy of 20 kcal/mol and increases linearly with D_2/CO ratio. The higher activation energy for chain termination than that for chain propagation explains the observed decrease in probability of chain growth, α, with increasing temperature.

The surface coverages by various carbonaceous species have also been determined from an analysis of the fitted transient response curves. It is concluded that the dominant species are monomeric building units, which occupy 0.2 to 0.6 ML. Growing alkyl chains, the direct precursors to hydrocarbon products occupy $\leq$ 0.2 ML. Adsorbed CO occupies an additional $\approx$ 0.7 ML. These estimates are found to be in good agreement with independent measurements made by temperature-programmed reaction spectroscopy.

APPENDIX

The apparent rate constant for propagation, $k_p\theta_m$, and the rate constant for termination, k_t, are calculated from α and τ using the following equations:

$$\alpha = \frac{k_p\,\theta_m}{k_p\,\theta_m + k_t} \qquad \text{(A1)}$$

and

$$\tau = \frac{1}{k_p\,\theta_m + k_t} \qquad \text{(A2)}$$

The value of τ is determined from a fit of eqn. 12 to the observed curves of $F_n(t)$.

The monomer coverage, θ_m, is calculated as follows. At steady-state, the rate of CO consumption must equal the rate at which monomeric species are consumed. This equality can be written as:

$$N_{CO} = \frac{\theta_m}{\tau_m} \qquad \text{(A3)}$$

Since N_{CO} is known from steady-state rate measurements and τ_m is known from the fit of eqn. 12 to the observed curves $F_n(t)$, θ_m can be evaluated. The value of k_p can be determined from eqn. A2.

The value of k_i can be calculated by recognizing that the rate at which carbon atoms are consumed from the C_1 pool is equal to the rate at which carbon atoms enter this pool. This leads to eqn. A4.

$$\frac{\theta_1}{\tau} = k_i\theta_m \qquad \text{(A4)}$$

Since

$$\frac{\theta_1}{\tau} = \frac{\frac{N_{C_1}}{k_t}}{\frac{1}{k_p\,\theta_m + k_t}} = \frac{N_{C_1}}{1-\alpha}, \qquad \text{(A5)}$$

k_i can be calculated from eqns. A4 and A5, using the measured values of N_{C_1} and α, and the previously determined value of θ_m.

The coverages by alkyl species can be calculated from the hydrocarbon turnover frequencies and the value of k_t. Since

$$N_{Cn} = k_t \theta_n \qquad \text{(A6)}$$

then

$$\sum_{n=1}^{n_1} \theta_n = \frac{1}{k_t} \sum_{n=1}^{n_1} N_{C_n} \qquad \text{(A7)}$$

REFERENCES

1. Anderson, R. B., 'The Fischer-Tropsch Synthesis', Academic Press, New York, 1984.
2. Biloen, P. and Sachtler, W. M. H., *Adv. Catal.* **30**, 165 (1981).
3. Kellner, C. S., and Bell, A. T., *J. Catal.* **70**, 418 (1981).
4. Dry, M. E., *Catal.-Sci. and Tech.* **1**, 150 (1981).
5. Dry, M. E., *Catal. Today* **6**, 183 (1990).
6. Bell, A. T., *Catal. Rev.-Sci. Eng.* **23**, 203 (1981).
7. Winslow, P., and Bell, A. T., *J. Catal.* **86**, 158 (1984).
8. Winslow, P., and Bell, A. T., *J. Catal.* **91**, 142 (19845).
9. Biloen, P., Helle, J. N., and Sachtler, W. M. H., *J. Catal.* **58**, 95

(1979).

10. Bonzel, H. P., and Krebs, H. J., *Surf. Sci.* **91**, 499 (1980).

11. Kobori, Y., Yamasaki, H., Naito, S., Onishi, T., and Tamaru, K., *J. Chem. Soc., Far. Trans. i* **78**, 1473 (1982).

12. Biloen, P., Helle, J. N., van den Berg, F. G. A., and Sachtler, W. M. H., *J. Catal.* **81**, 450 (1983).

13. Orita, H., Naito, S., and Tamaru, K., *J. Catal.* **90**, 183 (1984).

14. Zhang, X., and Biloen, P., *J. Catal.* **98**, 468 (1986).

15. Mims, C. A., and McCandlish, L. E., *J. Am. Chem. Soc* **107**, 696 (1985).

16. Mims, C. A., and McCandlish, L. E., *J. Phys. Chem.* **91**, 929 (1987).

17. Stockwell, D. M., and Bennett, C. O., *J. Catal.* **110**, 354 (1988).

18. Stockwell, D. M., Bianchi, D., and Bennett, C. O., *J. Catal.* **113**, 13 (1988).

19. Biloen, P., *J. Molec. Catal.* **21**, 17 (1983).

20. Bennett, C. O., in "Catalysis under Transient Conditions", eds: A. T. Bell and L. L. Hegedus, ACS Symposium Series **178**, 1 (1982).

21. Yokomizo, G. H., and Bell, A. T., *J. Catal.* **119**, 467 (1989).

22. Vannice, M. A., *J. Catal.* **37**, 449 (1975).

23. Vannice, M. A., and Garten, R. L., *J. Catal.* **56**, 236 (1979).

24. Vannice, M. A., and Sudhakar, C., *J. Phys. Chem.* **88**, 2429 (1984).

25. Reuel, R. C., and Bartholomew, C. H., *J. Catal.* **85**, 78 (1984).

26. Rieck, J. S., and Bell, A. T., *J. Catal.* **99**, 262 (1986).

27. Krishna, K. R. and Bell, A. T., *J. Catal.* **130**, 597 (1991).

28. Sano, M., Yotsui, Y., Abe, H., and Sasaki, S., *J. Biomed. Mass Spec.* **3**, 1 (1976).

29. Matthews, D. E., and Hayes, J. M., *Analyt. Chem.* **50**, 1465 (1978).

30. Jordan, D. S., and Bell, A. T., *J. Phys. Chem.* **90**, 4797 (1986).

31. Jenson, V. G., and Jeffreys, G. V., "Mathematical Methods in Chemical Engineering", Second Edition, Academic Press, 1977.

32. Dictor, R. A., and Bell, A. T., *Appl. Catal.* **20**, 145 (1986).

33. Schulz, H., *C_1 Molec. Chem.* **1**, 231 (1985).

34. Froment, G. F., and Hosten, L. H., *Catal.-Sci. and Tech.* **2**, 97 (1981).

35. Box, G. E. P., Hunter, W. G., and Hunter, J. S., "Statistics for Experimenters", John Wiley and Sons, 1978.

36. Shustorovich, E. and Bell, A. T., *Surf. Sci.* **248**, 359 (1991).

Chapter V

The Effect of Ethylene Addition on Fischer-Tropsch Synthesis

ABSTRACT

The C_2 products formed over Ru catalysts during Fischer-Tropsch synthesis often lie well below the Anderson-Schulz-Flory line describing the C_{4+} products. This has lead to speculation that either the surface precursor to these products plays a special role in chain growth, or that the ethylene formed re-adsorbs and re-enters the chain growth process. In this study, the effect of adding small amounts of ethylene to the CO/H_2 feed is investigated, using $^{13}CO/H_2$ and $^{12}C_2H_4$ to differentiate between the carbon sources. Ethylene addition suppresses methanation and results in an increase in the C_{3+} hydrocarbon formation rates. Isotopic tracer studies show that ethylene-derived carbon is present in the C_{3-8} product. Ethylene acts as an effective chain initiator, and ethylene-derived carbon accounts for 45% of a C_1 monomer species.

INTRODUCTION

The C_{3+} products observed during Fischer-Tropsch synthesis (FTS) are often characterized by an Anderson-Schulz-Flory (ASF) distribution. The concentration of methane usually lies above an extrapolation of the plot to lower carbon numbers, and the concentrations of C_2 and C_3 often lie below this line. The deviation of C_2 products from the ASF distributuion is particularly noticeable for FTS over Ru catalysts (1, 2). Two possible explanations have been proposed. The first is that the precursor to C_2 products may be characterized by a longer average lifetime than the precursors to higher molecular products (3,). The second possible explanation is that the ethylene, once formed, is readsorbed and reenters into the chain growth process (5, 6).

The first of these possibilities, the presence of a long lived C_2 intermediate during CO hydrogenation over Ru, has been suggested by Mims and co-workers (3, 4). Mims *et al.* (3) performed isotopic tracer experiments in which an abrupt switch was made from $^{12}CO/H_2$ to $^{13}CO/H_2$ in the reactor feed. NMR analysis of C_2 to C_5 olefin products allowed the determination of the fraction of ^{13}C at each position in these products. These studies showed that there are two distinct isotope replacement rates, with the two carbons at the aliphatic end having a distinctly longer average residence time than those in the remaining positions. Based on this evidence, the authors suggested the presence of two-carbon surface species that has a comparitively long residence time relative to higher molecular

weight surface species. The authors further proposed that these two-carbon species play the role of chain initiators. In a subsequent study, Mims *et al.* (4) proposed that these results could be explained by a model based on the assumption that initiators for chain growth spent over half their surface residence time as C_2 groups.

In situ addition of 1.2 % $C_2H_5NO_2$ to the CO/H_2 feed over a Ru/SiO_2 catalyst by Cavalcanti *et al.* (7) was performed to understand the role of the C_2 intermediate during FTS. The rate and selectivity for the C_3 through C_6 hydrocarbon products increased by 25% or more, while methane formation was reduced. An order of magnitude increase was seen in the C_2 and C_3 olefin to paraffin ratios. The addition of 1.2 % nitroethane did not alter the chain growth probability, which led the authors to speculate that the rates of chain initiation and propagation were both increased by comparable amounts when $C_2H_5NO_2$ was added. The authors concluded that both one-carbon and two-carbon units were formed, and that C_2H_x groups derived from $C_2H_5NO_2$ play an important role in enhancing chain initiation, while CH_x groups formed by hydrogenolysis participate in chain growth.

The effect of ethylene addition to CO/H_2 during FTS has been examined by a large number of investigators. Many of the older studies were conducted on Co and Fe catalysts and are reviewed by Eidus (8) and by Jordan and Bell (2). Over Co catalysts, added olefins readily incorporate into the FTS products and cause an increase in higher hydrocarbon production. Olefin readsorption is found to initiate hydrocarbon chains, the effectiveness of this process

decreasing rapidly with increasing chain length. Olefins also decompose to one-carbon fragments which then serve as monomer for chain growth. Over Fe catalysts, most researchers reported an increase in the production of higher molecular weight hydrocarbon products when olefins were added to the feed. Ethylene was also reported to act as a chain initiator.

Several recent studies of the effects of ethylene addition have been carried out over Fe and Co catalysts and these are briefly detailed here. Snel and Espinoza (9) have carried out ethylene addition experiments using an Fe catalyst. Their results indicated that the addition of 5 to 10 % ethylene to the feed readily initiates chain growth without altering the chain growth probability. The olefin selectivity increases, while the rate of hydrocarbon synthesis virtually doubles. A substantial fraction of the ethylene was incorporated into the C_3 products, whereas the amount of methane formed decreased. The authors also concluded that propagation by ethylene was unlikely. The incorporation of a small amount of ^{14}C-labelled ethylene during FTS over a doubly-promoted Fe catalyst was investigated by Tau *et al.* (10). 40% of the labelled ethylene was converted to ethane. The C_{5+} products contained ≈ 10% of the ethylene. With increasing carbon number, a decrease in radioactivity/mole was observed. The authors concluded that 85 % of the incorporated ethylene initiates chain growth.

Percy and Walter (11) used isotopic labelling and NMR spectroscopy to study the effects of adding 2 % ethylene to the synthesis feed passed over a Co/Al catalyst. From an analysis of the

propene, it was concluded that some ethylene dissociated to C_1 fragments. Ethylene incorporation as a C_2 unit was also observed. Adesina and co-workers (12) studied the effect of adding 1-2 % ethylene to different feed CO/H_2 mixtures over a commercial Co catalyst. The rates of formations of C_3-C_7 hydrocarbons increased from 50 to 100 %. The rate of methane formation and the probability of chain growth, α, remained unchanged. The effect of C_2H_4 addition was largest for the C_3 products, and decreased progressively for the higher hydrocarbons. The conclusion was that ethylene acted as a chain initiator.

In the case of FTS over Ru catalysts, it has been postulated that the low yields of C_2 and C_3 olefins may be due to the reincorporation of these products into growing chains (1). Studies by Kellner and Bell (1) showed that at concentrations above 1 %, ethylene addition enhanced the C_3 and C_4 product formation but suppressed the synthesis of C_{6+} hydrocarbons. Kobori *et al.* (13) have examined the effect of adding ^{12}C-labelled olefins to $^{13}CO/H_2$ mixtures, using GC-MS product analysis. The addition of ethylene in a 1:1 ratio to the CO resulted in an increase in the C_3 and C_4 products, with C_3 showing the larger increase. The methanation rate decreased. Propylene addition caused an increase in the C_2 and C_4 product, with a concurrent drastic reduction in methane formation. Isotopic distributions for C_1, C_3-C_5 alkanes indicate extensive ^{12}C incorporation. When $^{12}C_2H_4$ was added, 59 % of the methane was ^{12}C-labelled. More than 50 % of the C_3-C_5 alkanes consisted of only ^{12}C-

labelled molecules, whereas the percentage of products that were all ^{13}C-labelled was 4, 2 and 7 for C_3, C_4 and C_5 alkanes, respectively. ^{12}C-labelled propylene and octene addition also resulted in a large fraction of the C_1-C_5 alkanes containing ^{12}C. The authors concluded that carbon from the olefins can randomly incorporate into the reaction products. Morris *et al.* (14) studied the effects of ethylene and propylene addition on FTS over Ru supported on silica, 13x zeolite, titania and magnesia. For the silica and zeolite supported Ru, ethylene addition markedly enhanced the higher hydrocarbon formation rates without greatly influencing the mehanation rate, whereas for Ru/TiO_2 and Ru/MgO_2, the rate of higher hydrocarbons was enhanced by a factor of less than two and the methanation rate was reduced.

Jordan and Bell (2) have studied the interactions of ethylene with H_2 and CO over a Ru/SiO_2 catalyst, using isotopic labelling of the CO to differentiate carbon sources. These authors concluded that CO hydrogenation was strongly influenced by the presence of ethylene. The rates of C_3-C_6 hydrocarbon products increased, revealing maxima as the partial pressure of ethylene was increased. The methanation rate decreased when ethylene was added. With increasing partial pressure of ethylene, CO hydrogenation to hydrocarbons was progressively suppressed, while the hydroformylation of ethylene to propanal was enhanced. The product distribution could be described in terms of C_1 and C_2 monomer units that participate in chain propagation. The authors conclude that C_2H_4 is a more efficient source of intermediates for chain initiation

and chain growth than CO.

Mims and coworkers (15) have investigated the effects of adding ethylene, 1-hexene and 1-octene on FTS over Ru/ γ-Al_2O_3. Concentrations of upto 5 % ^{12}C- labelled olefin were added to H_2-^{13}CO mixtures. From an analysis of the products by GC-MS and NMR spectroscopy, it was established that alkenes could initiate the growth of higher molecular weight hydrocarbons, and with a lower probability, depolymerize to form lower molecular weight hydrocarbons. With added alkene concentrations of less than 1 %, the product distribution was substantially altered. The production of all hydrocarbon products except methane was increased. The increase in rates was highest for carbon numbers adjacent to the added alkene, and decreased for progressively higher and lower carbon numbers. NMR analysis of the products showed the presence of a C_2 initiator at the alkyl end of the 1-alkene products when ethylene was co-fed. The fractional ^{12}C-labelling of the last two positions of the 1-alkenes was $\approx$ 75%. These results were attributed to chain initiation from adsorbed ethylene and further growth from a common C_1 pool. About 54 % of the carbon in the methane formed when ethylene was added to the feed derived from the ethylene.

The objective of this study was to investigate the effect of adding small amounts of ethylene to the CO/H_2 feed mixture, using labelled ^{13}CO and $^{12}C_2H_4$ in order to differentiate between the carbon sources. Of particular interest was the determination of the extent to which chain initiation and propagation are affected by the added

ethylene.

EXPERIMENTAL

A 3.3% Ru/TiO_2 (Degussa P25) catalyst was used. Catalyst preparation and characterization have been described in Chapter 4. 1.35 gm of catalyst was loaded into a quartz microreactor. The product sampling and analysis system used were also identical to that described in Chapter 4.

Addition of ethylene to the feed were accomplished by using mixtures of 420 ppm ethylene in He (Matheson Gas) or 2 % ethylene in He (Matheson Gas). All experiments were conducted at 1 atm, 463 K and a H_2/CO ratio of 3. Total flow rate to the reactor was 100 cm^3/min comprised of 30 cm^3/min of H_2, 10 cm^3/min of CO and He or (He+ethylene) making up the remaining 60 cm^3/min.

RESULTS

After 20 min of reaction in 10:30:60 mixture of $CO:H_2:He$, the He in the feed was abruptly replaced by a He/420 ppm ethylene mixture. This resulted in the effective addition of 252 ppm of ethylene to the reactor feed. This concentration is similar to the amount predicted for C_2 by an ASF line based on the C_{4+} products. This small amount of ethylene caused almost no change in the rate of hydrocarbon production, as seen in Figure 1. There was also no

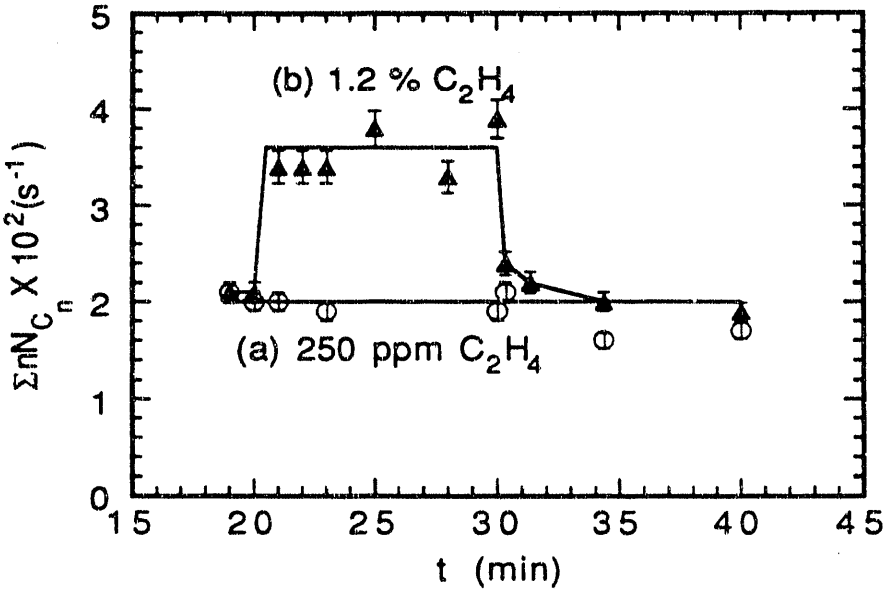

Fig. 1. Effect on total carbon in hydrocarbon product when ethylene is added to the CO/H_2 feed; (a) 250 ppm C_2H_4. (b) 1.2 % C_2H_4. Reaction conditions: T= 463 K; H_2/CO = 3. Ethylene is added from t = 20 min to t = 30 min.

change in the chain growth probability, α, which remained at 0.80 ± .02. Product analysis indicates that the C_2 product rose by a level equal to 32 % of the added ethylene. 68% of the added ethylene incorporated into products other than ethane, accounting for 2 % of these products. The rates of formation of the C_1 and C_{3+} products were unchanged. When 1.2 % ethylene was similarly introduced after 20 min of reaction, an immediate effect was seen (Fig. 1). The total rate of hydrocarbon production rose by a factor of 1.5. This rate is calculated as $\Sigma n N_{Cn}$, where N_{Cn} is the turnover frequency of the total hydrocarbon product of carbon number n. Figure 2 shows that the addition of 1.2 % C_2H_4 caused a small decrease in α to of 0.75 ± 0.2. After 30 min, a switch was made in the He stream back to pure He, and the activity dropped to the initial level in $CO/H_2/He$ and α rose to 0.8. Some deactivation was seen after 40 min on-stream consistent with the deactivation rate estimated for this catalyst in Chapter 2.

Experiments were also conducted in which reaction was started up in $CO/H_2/He/1.2$ % C_2H_4 and samples were taken between 20 and 40 min of reaction; these were then compared with reaction in the absence of ethylene. The results obtained confirmed the activity and α changes observed in the step change experiments shown in Fig. 1.

Of the ethylene fed, 12 % was converted to ethane and 82 % to other products. From these figures, it is determined that 68 % of the total hydrocarbon product, exclusive of C_2 products, derives from C_2H_4. The addition of 1.2 % C_2H_4 to the feed caused the CO

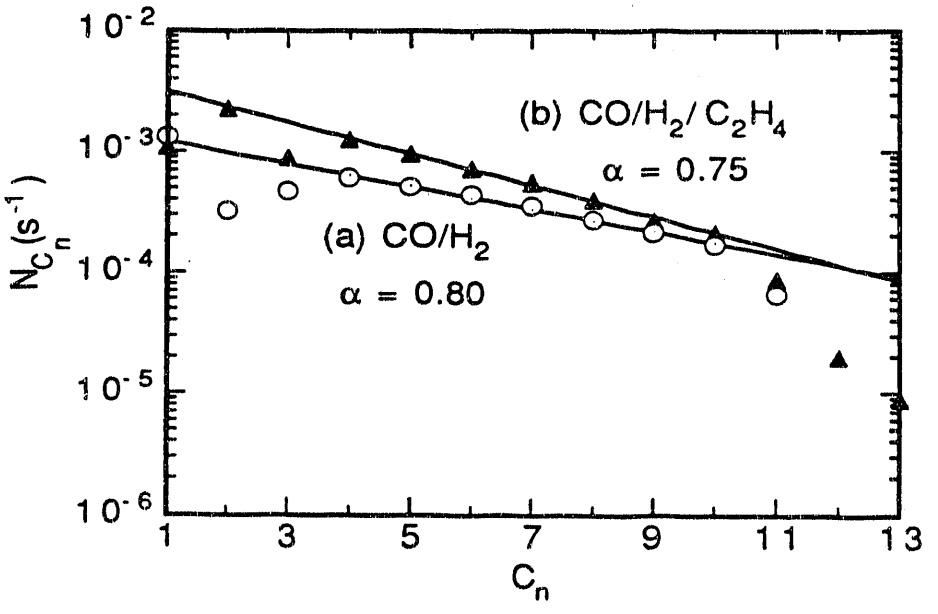

Fig. 2. Anderson-Schulz-Flory plot of N_{Cn} versus carbon number, C_n, in a) CO/H_2; (b) $CO/H_2/C_2H_4$. Reaction conditions: T= 463 K; H_2/CO = 3.

conversion to decrease from 20 % to 9 %.

The methanation rate decreased by 30 % upon C_2H_4 addition. Activity increases were observed for other products also, as can be seen from Figs. 2 and 3. Ethylene addition also resulted in an increase in the olefin content of the products. Figure 4 shows the olefin/paraffin ratio in the absence and presence of ethylene. There was no change in the identity of the reaction products formed when ethylene was added to the feed.

Experiments were also conducted in which 1.4 % C_2H_4 was reacted with 30 % H_2 in the absence of CO. The ethylene was completely converted since no ethylene was detected by gas chromatographic analysis of the reactor effluent. 54% of the ethylene was converted to methane, 43 % to ethane and 3 % to higher hydrocarbons upto C_{14}, which were detected in trace amounts.

To determine the extent to which the carbon derived from ethylene enters into the formation of hydrocarbon products, isotopic labelling was used. Reaction was initiated in $^{12}CO/H_2/He/1.2$ % C_2H_4 and after 20 min, the CO in the feed was replaced by ^{13}CO. After 10 min in labelled CO, the CO in the feed was switched back to ^{12}CO. Isotope-ratio gas chromatography (16, 17, 2) was used to study the incorporation of the ^{13}C from the CO into the hydrocarbon products from C_3-C_8. In order to assess the effect of the added ethylene, this experiment was repeated in the absence of ethylene in the He diluent stream.

Figure 5a shows the the transient response of the fraction of each product that is ^{13}C- labelled, F_n (n=3-8), when the feed is

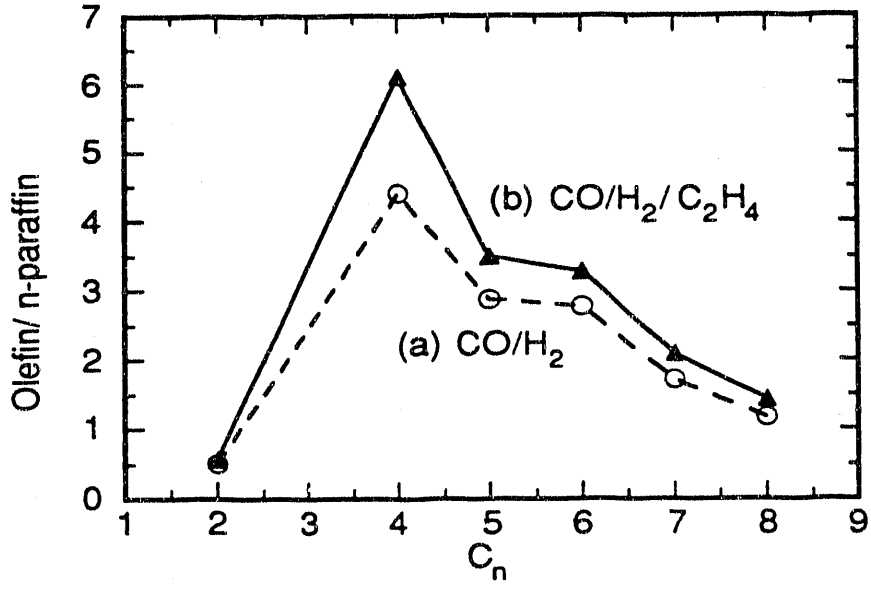

Fig. 4. Olefin/n-paraffin ratio as a function of carbon number, C_n in (a) CO/H_2; (b) $CO/H_2/C_2H_4$. Reaction conditions: T= 463 K; H_2/CO = 3.

changed from $^{12}CO/H_2/He$ to $^{13}CO/H_2/He$. These traces are similar to those presented in Chapter 4 and are representative of a sequential incorporation of ^{13}C into the products. Figure 5b shows the corresponding transient responses when 1.2% ethylene is present in the feed. Due to the presence of the $^{12}C_2H_4$, the products are never completely ^{13}C labelled, and consequently, the steady-state value of the ^{13}C fraction after 10 min in $^{13}CO/H_2/He/^{12}C_2H_4$ is used to normalize the fractions.

Table 1 lists the fraction of ^{12}C-labelled carbon in the C_3-C_7 products at steady-state reaction in $^{13}CO/H_2/He/^{12}C_2H_4$ (i.e., after 10 min of reaction). ^{12}C-labelled ethylene and ethane were also detected. These products were virtually 100% ^{12}C labelled because of the high concentrations of $^{12}C_2H_4$ in the feed.

DISCUSSION

Ethylene added to the CO/H_2 feed can facilely incorporate into the chain growth process; 68 % of the ethylene was incorporated into the non-C_2 hydrocarbon product when 250 ppm of ethylene was added to the CO/H_2 feed, and 82 % of the added ethylene was incorporated into C_1 and C_{3+} products when 1.2% ethylene was added. At the lower concentration, ethylene does not affect the rates of C_1 and C_{3+} product formation, but only displaces 2 % of the ^{13}CO. At a level of 1.2 % C_2H_4, the overall rate of hydrocarbon product formation rose by 50 % (Fig. 1). This increase is similar to that

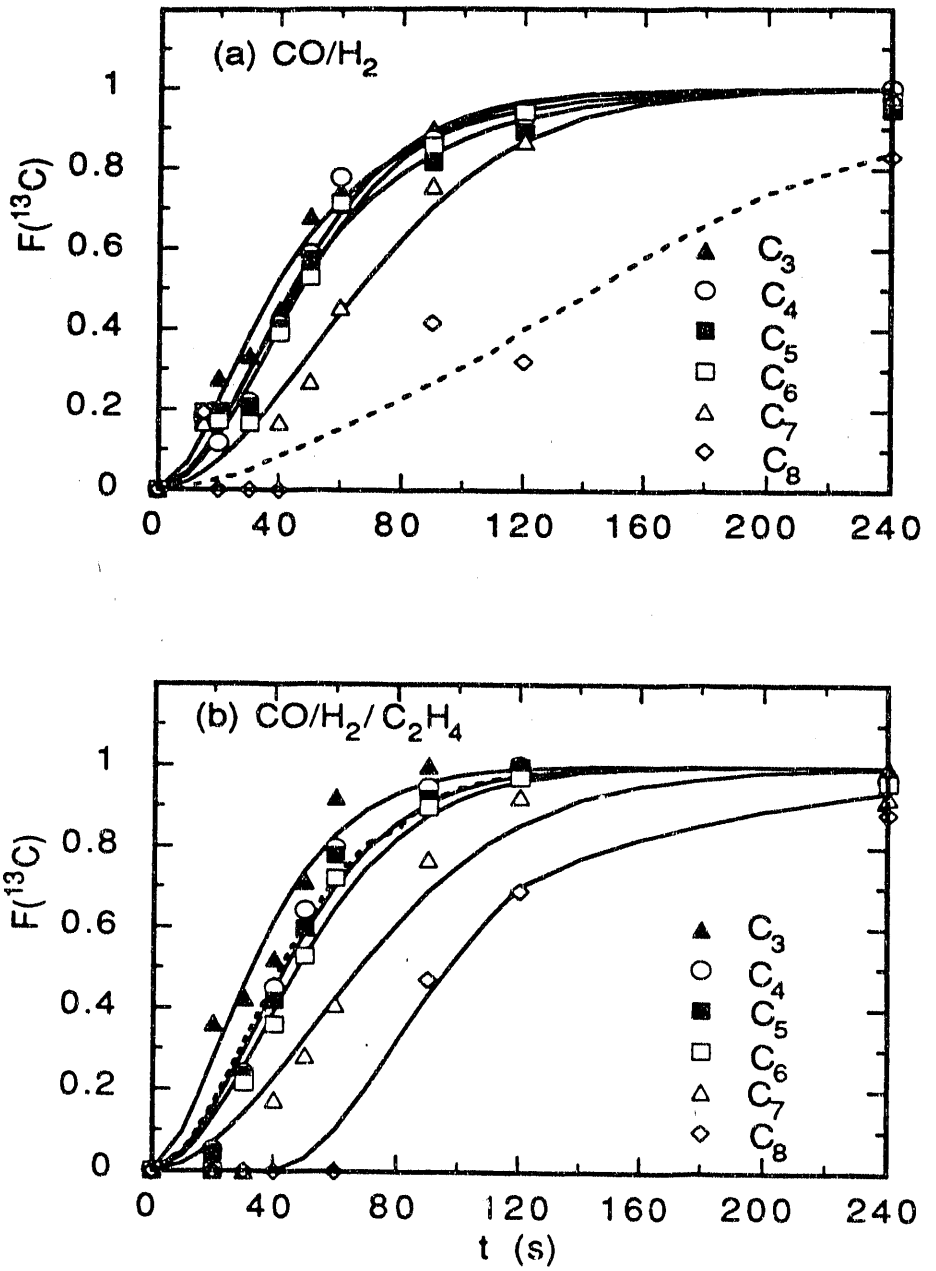

Fig. 5. $F(^{13}C)$ rise in the C_3-C_8 products; transient response after (a) a switch from $^{12}CO/H_2$ to $^{13}CO/H_2$ at t = 0; (b) a switch from $^{12}CO/H_2/^{12}C_2H_4$ to $^{13}CO/H_2/^{12}C_2H_4$ at t = 0. Reaction conditions: T= 463 K; H_2/CO = 3.

TABLE 1

Fraction ^{12}C in product at steady state in a
1.2 % $^{12}C_2H_4$/10 % ^{13}CO/30 % H_2/He mixture

Carbon Number	$F^{12}C$ (expt)	$F^{12}C$ (C_2 initiator)	$F^{12}C$ (C_2 initiator + 0.45 $^{12}C_1$)
3	0.79	0.66	0.82
4	0.77	0.50	0.73
5	0.68	0.40	0.67
6	0.65	0.33	0.63
7	0.64	0.29	0.61
8	0.57	0.25	0.53

observed by other researchers upon ethylene addition (1, 2, 8, 9, 12-15). Conversion of CO to products is reduced from 20 % to 9 % in the presence of 1.2% ethylene (i.e., ethylene displaces 55 % of the ^{13}CO) in a manner similar to that observed by Jordan and Bell (2), and CO accounts for only 32 % of the hydrocarbon product. The α value fell slightly on ethylene addition, (Fig. 2), in contrast to other reports (2, 9, 15). As can be seen from Figs. 2 and 3, this is due to the activity increase on ethylene addition being most pronounced for the C_3 and C_4 products and progressively decreasing for longer chains. This progressive decline in enhancement for the higher hydrocarbons was also noted by Mims *et al.* (15) .

An increase in olefin to paraffin ratio (shown in Fig. 4) on 1.2 % C_2H_4 addition is also in agreement with observations made by Snel and Espinoza (9). A concurrent decrease in methanation is seen here and has also been reported by other researchers (2, 9, 13-15). A probable cause is a lower hydrogen coverage on the surface that results in a reduced hydrogenation capability of the catalyst in the presence of ethylene, which would account for both these changes in product selectivity.

The isotopic transients presented in Fig. 5 show that the dynamics of ^{13}C incorporation into the products is not altered by the presence of ethylene. This indicates that the added ethylene does not alter the mechanism of chain propagation and termination.

The results shown in Table 1 indicate that in the presence of 1.2% C_2H_4, a large fraction of the products are ^{12}C-labelled. If the added ethylene only contributes $^{12}C_1$ units to the monomer pool, the

fraction of ^{12}C in all the products should be the same, at a level that reflects the fraction in this pool. The progressive decrease in the fraction of ^{12}C carbon suggests that adsorbed ethylene acts primarily as a chain initiator. If this were the only role of the added $^{12}C_2H_4$, then the fraction of ^{12}C would be 66 % in the C_3 products and the ^{12}C fraction would decrease rapidly in the higher molecular weight products with increasing number of carbon atoms. Table 1 shows the results of this calculation. Comparison with the experimental results reveals that the observed fractions are higher than these estimates and decrease much more gradually.

A slower decrease in the fraction of ^{12}C-labelled products would occur, if, in addition to acting as a chain initiator, $^{12}C_2H_4$ contributes to the pool of C_1 monomer units (e. g., $CH_{2,s}$). It can be assumed that a fraction of the C_1 monomer pool is ^{12}C-labelled, f_m, and further, that there is a constant fraction of C_2 initiators in the C_{3+} products, f_i. This would lead to the overall fraction of ^{12}C in a product of carbon number n, $F_n(^{12}C)$ to be:

$$F_n(^{12}C) = \frac{1}{n} [2f_i + (n-2)f_m]$$

A multiple-regression of this equation in two variables over the data for F_n (n = 3-8) yields values of $f_i = 1.03$ and $f_m = 0.45$. This indicates that the experimental results are consistent with every product molecule containing a $^{12}C_2$ initiator unit, and furthermore, that 45 % of the monomer is ^{12}C-labelled. Fixing this as the isotopic composition of the monomer pool, one can calculate the isotopic composition of the C_3-C_8 products. The results shown in Table 1

indicate that the calculated isotopic compositions agree reasonably well with those observed.

The results in Table 1 can be compared with those of Kobori *et al.* (13) and Mims *et al.* (15). Kobori and co-workers (13) report that when $^{12}C_2H_4$ was added to $^{13}CO/H_2$ at a level of 0.79:1:2, 59 % of the methane was ^{12}C-labelled. $^{13}C_i$ fractions (fraction of molecules containing i ^{13}C atoms) are reported for C_3-C_5 alkanes. These data show that 81 % or more of the C_{3-5} molecules contain a $^{12}C_2$ unit. Calculations from Kobori *et al.*'s data show that the total fraction of ^{12}C-labelled carbon is 85 % in the C_3 product, 88 % in the C_4 product and 69 % in the C_5 product. These data are also consistent with a model assuming extensive $^{12}C_2$ initiation and a monomer pool that contains 50-60 % ^{12}C. The 59 % fraction of this pool that is indicated by the isotopic composition of the methane is probably higher than that of 45 % indicated by our study due to the higher ratio of ethylene to CO used here. Mims *et al.* (15) report that when less than 5 % ethylene was added to the CO/H_2 feed, the methane contained $\approx$ 55 % ^{12}C, and that the fractional ^{12}C-labelling of the last two positions in the C_3 and C_4 1-olefins was 75%. These results can be compared to the 45 % ^{12}C-labelled monomer pool, and the 100 % $^{12}C_2$ initiation estimated here.

Jordan and Bell (2) have proposed that C_2 species can participate in chain propagation. If C_2 species were more efficient chain propagation units than C_1 species, the products with an even number of carbon units would probably contain more ^{12}C than the adjacent odd number products. Also, the incorporation of the ^{13}C

from CO would be slower in the even number products. The data presented in Table 1 indicates that the fraction of ^{12}C falls off with increasing carbon number, and the data shown in Fig. 5 show that ^{13}C incorporates successively into each carbon number product from C_3 upwards. This leads to the conclusion that C_2 species are not more effective chain propagation units that C_1 species.

The data in Fig. 3 indicate that the maximum activity increase is seen for the C_4 product, in contrast to the results of Adesina *et al.* (12) and Mims *et al.* (15) who reported a maximum increase in the C_3 product. Data by Jordan and Bell (2) indicate that the rate of production of the C_4 hydrocarbons can exceed that of the C_3 hydrocarbons for a range of reaction conditions used. Also, C_4 is the dominant product when ethylene is present in the feed (see Fig. 2), as has also been noted by Jordan and Bell (2). This could be due in part to ethylene homologation; calculations based on the observed isotopic composition, a $^{12}C_2$ initiator, a 45 % ^{12}C monomer pool and allowing for homologation indicate that roughly a fifth of the C_4 product may be formed by this process.

CONCLUSIONS

The effect of adding small amounts of ethylene to the CO/H_2 feed show that ethylene addition results in a substantial rate increase in hydrocarbon production, with the exception of methane. At an ethylene/CO ratio of 0.12, the ethylene accounts for 68 % of the hydrocarbon product. Isotopic tracer studies show that the

ethylene-derived carbon is present in the C_3-C_8 product, the fraction of such carbon decreasing as the hydrocarbon chain length increases. These observations lead to the conclusion that ethylene acts as an effective chain initiator, and as a source of carbon for C_1 monomer units. For the conditions of the present experiment, 100 % of the chain initiator and 45 % of C_1 species for chain propagation are derived from ethylene.

REFERENCES

1. Kellner, C. S., and Bell, A. T., *J. Catal.* **70**, 418 (1981).
2. Jordan, D. S., and Bell, A. T., *J. Phys. Chem.* **90**, 4797 (1986).
3. Mims, C. A., McCandlish, L. E., and Melchior, M. T., *Catal. Lett.* **1**, 121 (1988).
4. Mims, C. A., McCandlish, L. E., and Melchior, M. T., "Proceedings of the 9th International Congress of Catalysis", Vol IV, p1992 (1988).
5. Iglesia, E., Reyes, S. C., and Madon, R. J., *J. Catal.* **129**, 238 (1991).
6. Madon, R. J., Reyes, S. C., and Iglesia, E., *J. Phys. Chem.* **95**, 7795 (1991).
7. Cavalcanti, F. A. P., Oukaci, R., Wender, I., and Blackmond, D. G., *J. Catal.* **123**, 270 (1990).
8. Eidus, Ya. T., *Russ. Chem. Rev. (Engl. Transl.)* **36**, 338 (1967).
9. Snel, R., and Espinoza, R. L., *J. Molec. Catal.* **43**, 237 (1987).
10. Tau, L-M., Dabbagh, H. A., Chawla, B., and Davis, B. H., *Catal. Lett.*

7, 141 (1990).

11. Percy, L. T., and Walter, R. I., *J. Catal.* **121**, 228 (1990).
12. Adesina, A. A., Hudgins, R. R., and Silveston, P. L., *App. Catal.* **62**, 295 (1990).
13. Kobori, Y., Yamasaki, H., Naito, S., Onishi, T., Tamaru, K., *J. Chem. Soc, Faraday Trans. 1* **78**, 1473 (1982).
14. Morris, S. R., Hayes, R. B., Wells, P. B., Whyman, R., *J. Catal.* **96**, 23 (1985).
15. Mims, C. A., Krajewski, J. J., Rose, K. D., and Melchior, M. T., *Catal. Lett.* **7**, 119 (1990).
16. Sano, M., Yotsui, Y., Abe, H., and Sasaki, S., *J. Biomed. Mass Spec.* **3**, 1 (1976).
17. Matthews, D. E., and Hayes, J. M., *Analyt. Chem.* **50**, 1465 (1978).

Appendix I

Data Acquisition and Reduction

This appendix describes data acquisition and reduction for the different types of experiments used to generate data presented in this thesis. Listings of computer programs that have been used for this purpose are provided at the end of this Appendix. Primary gas chromatography (GC) and mass spectrometry (MS) data acquisition and reduction programs are written in 'Turbobasic' (Borland International Inc., Scotts Valley, CA). Lotus 123 (Lotus Development Corporation, Cambridge, MA) has been used for further data reduction. These have been used on an IBM-PC (XT) microcomputer.

1. Data Acquisition

Data can be recorded from the MS or from the GC alone; also, GC-MS data where both data are followed (and are linked) can be followed. Three main data acquisition programs are used. The first, called "LOCTMS", is used to locate digital/analog (D/A) mass spectrometric peak locations; these are then used in the other two programs. The second, called "MSONLY", records MS data continuously for a user-programmed duration. Up to 5 masses can be monitored; data is collected at 2 readings/s for each mass programmed. The third program is called "TSNEWGC" and records GC-MS data; due to the lengthy nature of these experiments, data is stored only if a GC or MS peak is found. Peak-sensing criteria are built-in for the GC and MS separately and can be user-modified. This

program is used for either GC data alone or GC-MS data. Data is collected on a GC channel and up to 5 MS channels at 2 readings/s. Detailed descriptions of these programs follow. All the programs are user-friendly and will prompt the operator to perform the next step at each stage.

1.1 LOCTMS

This program is used to search the mass spectrum for peak locations. The MS is first set up for computer control: 'Program' EXTernal, 'Function' MULTiplier and 'Gain' 6. The cursor keys now controls the external D/A location that sets the amu being scanned on the MS. The '6' key is coarse up, the '4' key is coarse down, the '8' key is fine up and the '2' key is fine down. An input of 'G' prompts the user to input the desired gain. Finally, the computer provides the desired D/A location after a peak is found.

1.2 MSONLY

MSONLY is used to generate on-line MS data. Initial input consists of a filename to save to on the 'B' drive, and the number of masses that are of interest, as well as the total length of the experiment (which is limited by the number of masses followed). Then on further prompting, the masses desired are programmed. A D/A location and external gain factor for each mass are required, and can be searched for, since 'LOCTMS' is built in to this program, or keyed in if already known. At this point, the input parameters are saved, and the computer display indicates its readiness to record data after a keystroke input prompt from the operator. This program is used for experiments such as temperature programmed surface reaction (TPSR; see Chapter 2 or 4) or on-line isotopic transients

(Chapter 4).

1.3 TSNEWGC

This program is used to acquire GC and GC-MS data. It is used for computer control of the 10-port GC 'load/inject' valve and the 16-sample loop Multiposition GC valve(MPV) and also to externally start up the Perkin-Elmer Sigma 3B GC and subsequently the MS in order to analyze all the samples stored in the MPV.

The first part of the input consists of MS parameters, which are entered in a manner identical to that used in 'MSONLY'. Then GC loop data is entered; the user keys in the number of loops to be filled, the load time for each of these loops and the total time required for a sample to elute through the GC column. (This depends on the GC temperature program used.) Also programmed is the delay to start recording MS data after GC start-up; this allows the big injection air wave to elute through the MS without being recorded or changing the He carrier baseline. At this point, the computer waits for a user keypad prompt to begin the experiment.

The GC-MS temperature program takes ≈ 30 min for each loop; there are a possible maximum of 16 such loops. So a large amount of data is generated as GC and MS readings versus time. Therefore, to simplify and limit data acquisition, the computer only stores data if a peak is detected in the GC or on any of the MS traces. Peak retention times and GC or MS reading vs. time data for 20 s around a peak are recorded. This is sufficient as each peak lasts about 6s. GC or MS peak criteria can be altered by the user in the program. MS criteria, which are more critical (since MS data is in general more noisy than the GC) can be changed by changing the input text file,

'mspkpar.txt'. This has 4 parameters, {n_1, n_2, n_3 and n_4}; n_1 is the minimum step size in numbers from 0-4096. A number 1 corresponds to a 2.44 mV change in the 0-10V MS output to the A/D converter. The value used here is 2. n_2 is the minimum number of consecutive rising steps, each equal to or greater than n_1. A value of 3 is used here. n_3 is the absolute peak height above the baseline and a value of 15 is used. Finally, n_4 sets the MS delay time between sampling to allow the MS readings to settle. This is currently set at 3000. Optimization of these parameters allows picking up almost all peaks that are visible on the screen, while minimizing the pick-up of random noise.

When the operator starts up the computer to begin the experiment, the computer initially 'homes' the 16-loop MPV and sets the 10-port valve to the LOAD position. It then switches the MPV to loop 2 and proceeds to fill loop 2 for the programmed loading time. Before it switches to loop 3, the computer beeps and awaits a user-prompt. This is to allow the operator to do the isotopic transient switching. After the loading has re-commenced, the computer loads the required number of loops for the programmed loading times. The last loop to be filled is always loop 1. After all loop-loading is complete, the computer awaits a user-prompt to commence sample-injection and GC-MS data acquisition. After this, it also waits for a GC-Ready signal from the GC. (Usually, it is ensured that when loading is complete, the GC is also ready for injection.)

When the GC is ready, the computer externally starts up the GC temperature program and immediately switches the 10-port GC

valve to the 'INJECT' position. Loop1 is the first loop to elute through the column. GC-MS data is now acquired for the input time/loop. After the sample has eluted, the IBM-PC writes the peak information to disk and again waits for the GC to get ready before switching the MPV to the next loop for injection. When saving the data for loop n, the computer also checks that there is sufficient space for the next loop data, and if found to be insufficient, a message appears on the screen, prompting the user to insert another disk and hit enter, before the next loop is injected. After all the loaded loops are eluted, the computer stops data acquisition. There is an error-handling Turbobasic subroutine built into TSNEWGC that writes into an error file if the program is abruptly terminated in the middle of an experiment. This aids trouble-shooting if the program fails.

2.0 Data Reduction

There are two programs which read data acquired by 'MSONLY' and 'TSNEWGC'. This data is then stored in files that are imported into LOTUS 123 worksheets, which are used for the next step in data analysis. Finally, if model fitting is required, a FORTRAN program, 'MINIMUM' is used.

2.1 Reading MS or GC-MS data

'MSONLY' data is read by a program called 'READMS'. This program will display on the screen the mass spectrum of any of the masses recorded; these can also be plotted on a Hewlett Packard plotter to provide a hard copy. There are two little cursors that appear on the screen that can be moved using the right and left

arrow cursor keys. These can be positioned where desired and if 'B' is entered, a new baseline will be drawn between them. If they are positioned and 'return' is entered, the area between the two cursors will be integrated and printed out on the screen. The average carrier He signal is also computed. The data for any mass can also be saved to disk as time-reading x-y pairs for further analysis by LOTUS. Calibrations are then used to convert the MS areas to actual amounts of products. MS data is usually normalized to the He signal to account for changes in the MS electron multiplier gain.

'TSNEWGC' data is read by a program called 'READTS'. This allows us to look at any loop. GC or MS data can be separately scanned, peak by peak, and peak areas can be integrated using the two screen cursors. This takes a lot of time when many peaks are present and so in both cases, the data can be automatically integrated using the 'Autointegrate' option. The most used feature of this program is the 'Autoalign' option; this will autointegrate the GC and every MS spectrum recorded for a given loop, then align the GC peaks to a library of known compound based on GC peak retention times, and finally align the MS peaks to the corresponding GC peak based on input GC-MS time delay for each known GC peak. The final step is a printout of all the data; integrated GC peaks and each MS channel and finally the aligned, integrated GC/MS data. The GC-MS data is saved to disk.

The GC- MS data is first examined for accurate peak identification and GC-MS peak line-up. (Since retention times are very sensitive to GC carrier flow rate and temperature program, there can be some variations.) Also, at low concentrations,

especially in the MS, peaks are not well-separated and require manual integration using the cursors. Any such corrections are noted on the GC-MS data hard copy. The next step in data reduction uses a LOTUS 123 spread sheet. Usually there may be up to 60-70 GC peaks and MS peaks corresponding to these on 4 masses for each sample loop, and a possible 16 such samples.

2.2 LOTUS 123 Data Analysis

The large quantity of data necessitates the use of three worksheets for data analysis. First, the GC-MS data are imported, loop by loop, into a LOTUS worksheet. This is the 'raw' data, and in this worksheet, blanks are inserted for missing peaks and any corrections in peak identification or areas is incorporated. Data for each loop consists of 5 columns; peak ID, peak number, GC area, and MS areas normalized by the He signal for ^{12} and ^{13} labelled compounds.

Once the 'GCMS.RAW' spreadsheet is created, GC data is analyzed in a separate spreadsheet. Here, GC areas are copied from the .RAW worksheet for each loop; calibrations for C_{1-14} species are used to convert these areas to concentrations. For each loop, the sum of all the concentrations for each carbon number is calculated; these sums are further used to obtain an ASF chain growth parameter for each loop. Finally, a calculation is made of the ΣnN_{Cn} product to calculate overall CO conversion for each loop.

The final worksheet is used for GC-MS data analysis. Here, for each loop, GC concentrations are imported from the GC worksheet;

MS data is copied from the .RAW worksheet. Here, MS calibrations are used to obtain ^{12} and ^{13}C MS peak concentrations. These are then divided by the corresponding GC peak concentration to obtain fractional isotopic concentrations. At a given carbon number, the MS concentrations of all detected peaks are added together and divided by the total concentrations of those peaks (obtained from GC data) for ^{12}C-labelled and ^{13}-C labelled isotopic fractions. MS data is often noisy, and there is also a 1% ^{12}CO impurity in the feed ^{13}CO; hence, further normalization is necessary. The steady state loop data in either isotope is used for this purpose. (Typically, loop 2 is the steady-state in $^{12}CO/H_2$ and loop 10 the steady-state in ^{13}CO/H2.) All data is converted to isotopic rise data, by using the fact that $F^{rise}+F^{fall} = 1.0$ at any time; for each time, the average of the ^{12}C and ^{13}C rise curve for each product is calculated. Finally, 16 points of t, $F^{rise}(t)$ are obtained. This data is now fitted to the chain growth model using a FORTRAN program, 'Minimum'. This is described in Appendix 3.

MSONLY.TXT

```
1 REM Turbo Basic program to run MS for Kamala ver 1.0 04-26-88
10 KEY OFF:CLS:REM turn off key line
12 PRINT "Switch valve interface to manual"
14 PRINT "Type any key to continue"
16 IF INKEY$="" THEN 16
20 PRINT "Place DATA disk into drive B:"
40 INPUT "Enter file name you want to save on ";FLNAME$
60 OPEN "b:"+FLNAME$+".cmy" FOR OUTPUT AS #2
70 PRINT "Enter comments, terminate with a blank line"
80 INPUT COMMENT$:IF COMMENT$="" THEN 92
90 PRINT#2,COMMENT$:GOTO 80
92 CLOSE 2
100 BASE1=&H300
110 HEXTABLE$="0123456789ABCDEF"
120 OUT BASE1+3,&H90:REM init Metrabyte PIO12
130 INJECTOR=BASE1+1:LOOPVALVE=BASE1+2:STATES=BASE1
140 LD=2:INJECT=4:FIDSTART=1:INJNORM=&H6
150 STP=&H80:HOME=&H40:SELPOS=&H20:LOOPNORM=&HFF
160 OUT INJECTOR,INJNORM
170 OUT LOOPVALVE,LOOPNORM
180 OPEN "com1:1200,n,7,1,rs,cs,ds,cd"AS 1:REM open channel of
Omega
200 BASE2=&HE6A0
210
ATOD=BASE2:DTOA=BASE2+4:COUNTER=BASE2+8:PLOTTER=BASE2+12
220
ADSTART=BASE2+&H1C:CTRESET=BASE2+&H1D:RDYCLR=BASE2+&H1E:
SCAN1=BASE2+&H1F
230 REM program 8255s of the GCMS interface
240 OUT ATOD+3,&H92:OUT DTOA+3,&H89:OUT COUNTER+3,&H92:OUT
PLOTTER+3,&H80
250 REM define dafault I/O condition
260 OUT COUNTER+2,16+4+1:REM select time-base 2 Hz
270 OUT DTOA+1,0:OUT DTOA,0:REM set mass program to 0
280 OUT PLOTTER+2,255:OUT PLOTTER+1,255:OUT PLOTTER,255:REM
zero  plotter
```

```
290 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
370 DEF FNFREESPACE(DRIVE%) 'find free space on drive
380 REG 4,DRIVE%            '0=default,1=A:,2=B:,etc.
390 REG 1,&H3600            'AH=function number
400 CALL INTERRUPT &H21     'DOS function call
410 FNFREESPACE=CSNG(REG(2))*REG(3)*REG(1)
420 END DEF
490 REM
------------------------------------------------------------------
------------
510 GOSUB 5000:REM get MS parameters
520 numm=nummass%
550 NEED=(NUMM*5000+1000)
552 IF NEED>FNFREESPACE(2) THEN PRINT "Not enough disk space
!!":END
562 PRINT
570 PRINT "Turn damping control of UTI to min."
580 PRINT "Select gain control of UTI to (-11)"
600 PRINT "Type any key when ready"
610 IF INKEY$="" THEN 610
620 REM DATA ACQ. START HERE-------------------------
622 MASSPG=MASSLOC(0):GOSUB 1000:REM move mass program to
1st mass
624 OUT ATOD+2,0+MASSGAIN%(0)*32:REM select 1st mass gain and
ch. 0
630 STARTDATE$=DATE$:STARTTIME$=TIME$:IF STARTDATE$<>DATE$
THEN 630
632 GOSUB 4000:REM save parameters
644 SCREEN 2:CLS:REM graphics screen
680 GOSUB 3000:REM data acquisition procedure
682 out atod+2,0+6*32:rem switch to ch.0 and gain 6
690 GOSUB 4200:REM data saving procedure
702 SCREEN 0:REM text screen again
710 FINISHDATE$=DATE$:FINISHTIME$=TIME$:IF FINISHDATE$<>DATE$
THEN 710
720 PRINT "experiment starts at ";STARTDATE$;" ";STARTTIME$
730 PRINT "          finishes at ";FINISHDATE$;" ";FINISHTIME$
740 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
```

```
990 END:REM end of main
program----------------------------------------------
1000  REM  output  dtoa  data----------------------------
1010 IF MASSPG<0 THEN MASSPG=0
1020 IF MASSPG>65535! THEN MASSPG=65535!
1030  MVAL=INT(MASSPG/256):LVAL=MASSPG-MVAL*256
1040 OUT DTOA+1,MVAL:OUT DTOA,LVAL
1050 RETURN
3000  REM data acquisition
procedure---------------------------------
3030  element%=0:x1=0:y1=199
3010 rdyfg=(INP(ATOD+1) AND &H20):REM WAIT UTIL COUNTER IS
READY
3013  if rdyfg=0  then goto 3010
3014 OUT RDYCLR,0:REM clear counter ready flag

3200  REM  read  masses---------------------
3202 if nummass%=0 then 3400:rem no mass
3210 FOR I=0 TO NUMMASS%-1
3220  REM mass is selected in previous loop
3260 OUT ADSTART,0:REM strobe A to D
3270 MASSREAD%=(INP(ATOD+1) AND 15)*256+INP(ATOD):REM read
mass  intensity
3280 MASSINT%(element%,I)=MASSREAD%:REM put reading into
buffer
3290 IF I=(NUMMASS%-1) THEN NEXTMASS%=0 ELSE NEXTMASS%=I+1
3300  out  atod+2,0+massgain%(nextmass%)*32:rem  select  new  gain
3310 MASSPG=MASSLOC(NEXTMASS%):GOSUB 1000:REM program
nextmass
3320  for xxx=0 to 20:next:rem delay gen
3380  NEXT  I:REM  NEXT  MASS---------------------
3390  y2=199-massread%/22:x2=x1+1
3400  line  (x1,y1)-(x2,y2)
3402  x1=x2:y1=y2
3404  if  x1>600  then  x1=0:y1=199:cls
3410 ELEMENT%=ELEMENT%+1
3412 LOCATE 23,1:PRINT "At point ";ELEMENT%
```

```
3430 if element% < numpoint% then 3010
3470 RETURN:REM data acq proedure
return----------------------------
4000  REM  Saving  parameters  routine--------------------------
4010 PRINT "Saving parameters; please wait"
4020 OPEN "b:"+FLNAME$+".paz" FOR OUTPUT AS #2
4040 PRINT#2,NUMMASS%
4042  print#2,numpoint%
4050 FOR I=0 TO NUMMASS%-1
4060 PRINT#2,MASSNUM%(I),MASSGAIN%(I)
4070 NEXT I
4140  CLOSE  2:RETURN:REM  ----------------
4200  REM  saving  data  -------------------
4202 LOCATE 1,1
4204  PRINT "Saving data, please wait"
4340 DEF SEG=VARSEG(MASSINT%(0,0))
4350 LENGTH=VARPTR(MASSINT%(numpoint%,NUMMASS%-1))
4360 BSAVE "b:"+FLNAME$+".mas",0,LENGTH
4590 RETURN:REM=========
5000  REM  get  MS  parameters  ---------------------------
5010 INPUT "Enter number of masses interested ";NUMMASS%
5011  maxpoint%=int(16000/nummass%):print "Max. # of points
";maxpoint%
5012 input "Enter number of data points per mass ";numpoint%
5013 if numpoint%>maxpoint% then print "out of range":goto 5012
5014 if nummass%=0 then 5300
5016 dim massint%(numpoint%,nummass%)
5017 dim
massnum%(nummass%),massloc(nummass%),massgain%(nummass%)
5020 PRINT "Locate mass :"
5030 PRINT "     to the prompt of 'LOCATE' use keypad key"
5040 PRINT "       '8' for fine up"
5050 PRINT "       '2' for fine down"
5060 PRINT "       '6' for coarse up"
5070 PRINT "       '4' for coarse down"
5080 PRINT "     to move mass around "
5082 PRINT "        'K'     to use key entry"
5084 PRINT "        'G'     to select gain"
```

```
5090 PRINT "     and 'ENTER' to select location":PRINT
5100 MASSPG=0:GOSUB 1000:REM preset to mass 0
5110 FOR I=0 TO NUMMASS%-1
5120 PRINT I+1;"  ";
5130 INPUT "enter mass number ";MASSNUM%(I)
5140 PRINT "LOCATE    ";
5150 TEMP$=INKEY$:IF TEMP$="" THEN 5150
5160 IF ASC(TEMP$)=13 THEN 5250:REM select location of mass
5162 if temp$="K" then 5242
5164 if temp$="G" then gosub 5400:rem change gain%
5170 IF LEN(TEMP$)<>2 THEN 5150
5180 TEMP=ASC(RIGHT$(TEMP$,1))
5190 IF TEMP=72 THEN MASSPG=MASSPG+1
5200 IF TEMP=80 THEN MASSPG=MASSPG-1
5210 IF TEMP=77 THEN MASSPG=MASSPG+100
5220 IF TEMP=75 THEN MASSPG=MASSPG-100
5230 GOSUB 1000:REM program D to A
5240 GOTO 5150
5242 input "Enter D to A setting ";MASSPG:gosub 1000
5250 PRINT "at D to A setting of ";MASSPG
5260 MASSLOC(I)=MASSPG
5270 MASSGAIN%(I)=gain%
5272 print "gain control for mass";massnum%(i);" is";massgain%(i)
5280 PRINT
5290 NEXT I
5300
RETURN:REM-----------------------------------------------
5400  rem  change  gain%  --------------------
5410 input "enter mass gain (0 to 7) ";gain%
5420 if (gain%<0) or (gain%>7) then 5410:rem out of range
5450 out atod+2,gain%*32:rem switch gain
5460   return:rem   -------------------------
```

READMS.TXT

```
100 REM loading data from MSONLY program ver 1.0 5-27-88 by
Henry Chan
110 base2=&he6a0
120  plotter=base2+12
130  out  plotter+3,&h80
140  xplot%=0:yplot%=0:gosub  4200:rem  zero  plotter
310 PRINT "Place data disk in drive B:, type any key to continue"
312 IF INKEY$="" THEN 312
314 FILES "b:*.paz"
320 INPUT "Enter file name ";FLNAME$
330 OPEN "b:"+FLNAME$+".cmy" FOR INPUT AS #2
340 WHILE NOT EOF(2)
350 LINE INPUT#2,COMMENT$
360 PRINT COMMENT$
370 WEND
380 CLOSE 2
390  INPUT "Is this the right one (y/n) ";YES$
400 IF YES$="y" THEN 430
410 IF YES$="n" THEN 310
420 GOTO 390
430 PRINT
500   REM   Loading   parameters------------------------
510 PRINT "Loading parameters, please wait"
520 OPEN "b:"+FLNAME$+".paz" FOR INPUT AS #2
540 INPUT#2,NUMMASS%:PRINT "Number of masses observed
=";NUMMASS%
542  input#2,numpoint%
544  dim
massint%(numpoint%,nummass%),massnum%(nummass%),massgain%(
nummass%)
546 dim ydel%(10)
550 FOR I=0 TO NUMMASS%-1
560 INPUT#2, MASSNUM%(I),MASSGAIN%(I)
570 PRINT "MASS";I+1;"is";MASSNUM%(I),"Gain factor";MASSGAIN%(I)
580 NEXT I
660  CLOSE  2:REM  parametes  loaded----------------
```

```
1000  REM  read  data----------
1100 print "reading data; please wait"
1110 def seg=varseg(massint%(0,0))
1120 bload "b:"+flname$+".mas",0
1180 screen 2
1190 rem data
loaded-----------------------------------------------
1200 rem display data
1210 for i=0 to nummass%-1
1220 print "mass";massnum%(i);" used gain factor";massgain%(i)
1230 next i
1240 input "Which mass (0 to exit)";massdisp%
1241 if massdisp%=0 then 2990
1250  msptr%=-1
1252 for i=0 to nummass%-1
1260 if massnum%(i)=massdisp% then msptr%=i
1262 if massnum%(i)=4 then refptr%=i
1270 next i
1280 if msptr%=-1 then print "mass not found; try again":goto 1200
1281 input "Save in x,y pair (y/n) ";an$
1282 if an$<>"y" then 1284  else gosub 4400
1283 input "Do you want to continue to plot (y/n) ";an$:if an$<>"y"
then 2980
1284 input "Plotter output (y/n) ";an$
1285 if an$="y" then plotyes=1 else if an$="n" then plotyes=0 else
goto 1282
1286 if plotyes then print "Plotting is selected"
1290 print "Calculating scale; please wait"
1300 ymax%=-1:ymin%=4096
1310 for i=0 to numpoint%-1 step int(1+numpoint%/600)
1320 y%=massint%(i,msptr%)
1330 if y%>ymax% then ymax%=y%:xmax%=i
1340 if y%<ymin% then ymin%=y%:xmin%=i
1350 next i
1352 print ymax%,xmax%,ymin%,xmin%
1360  yscale%=(ymax%-ymin%)/170+1
1362 print yscale%
1370  yoffset%=ymin%
```

```
1380 print yoffset%
1390 print "Resetting plotter, lift pen and type any key to continue"
1400 if inkey$="" then 1400
1410 xplot%=0:yplot%=0:gosub 4200:rem zero plotter
2000 print "Type any key to start plotting routine"
2004 if inkey$="" then 2004
2006 base1%=0:base2%=numpoint%-1:head%=base1%:tail%=base2%
2010 cls:gosub 3000:xplot=0
2090 xscale%=numpoint%/600+1
2094 intsts=0:integral=0:flag%=0
2096 x1=0:y1=190-(massint%(0,msptr%)-yoffset%)/yscale%
2098 if plotyes then inc=1 else inc=int(1+numpoint%/600)
2100 for i=0 to numpoint%-1 step inc
2110 x2=i/xscale%:y%=massint%(i,msptr%):yplot%=y%
2120 y2=190-(y%-yoffset%)/yscale%
2140 line (x1,y1)-(x2,y2)
2141 if plotyes then gosub 4200
2142 y1=y2:x1=x2:xplot=xplot+0.90:xplot%=int(xplot)
2290 if inkey$=chr$(27) then i=numpoint%:flag%=1
2300 next i
2302 if flag%=1 then 2980
2304 plotyes=0
2310 locate 2,1:print "Integration routine; use 'ESC' to return to
main menu "
2330 y1=190-(massint%(base1%,msptr%)-yoffset%)/yscale%
2340 y2=190-(massint%(base2%,msptr%)-yoffset%)/yscale%
2350 line (base1%/xscale%,y1)-(base2%/xscale%,y2):rem line base
line
2352 locate 4,1:print "BASE LINE " massint%(base1%,msptr%);" at
";base1%;" ";
2354 print massint%(base2%,msptr%);" at ";base2%;"                    ";
2360 y1=190-(massint%(head%,msptr%)-yoffset%)/yscale%
2370 y2=190-(massint%(tail%,msptr%)-yoffset%)/yscale%
2380 line (head%/xscale%,y1-10)-(head%/xscale%,y1-20)
2390 line (tail%/xscale%,y2-10)-(tail%/xscale%,y2-20)
2392 locate 5,1:print massint%(head%,msptr%); " at ";head%;"  ";
2394 print massint%(tail%,msptr%);" at ";tail%;                "
";
```

```
2400 ky$=inkey$:if ky$="" then 2400
2410 ky%=asc(ky$)
2420 if ky%=27 then 2980
2430 if ky%=13 then 2600
2440 if ky$="B" then gosub 2800:base1%=head%:base2%=tail%:goto
2330
2442 if ky$="R" then goto 2010:rem redraw
2445 if ky$="8" then if head%<numpoint%-21 then
nhead%=head%+20:goto 2510
2446 if ky$="2" then if head%>20 then nhead%=head%-20:goto 2510
2447 if ky$="6" then if tail%<numpoint%-21 then
ntail%=tail%+20:goto 2510
2448 if ky$="4" then if tail%>20 then ntail%=tail%-20:goto 2510
2450 if len(ky$)<2 then 2400
2460 temp=asc(right$(ky$,1))
2470 if temp=72 then if head%<numpoint%-1 then nhead%=head%+1
2480 if temp=80 then if head%>0 then nhead%=head%-1
2490 if temp=77 then if tail%<numpoint%-1 then ntail%=tail%+1
2500 if temp=75 then if tail%>0 then ntail%=tail%-1
2510 line (head%/xscale%,y1-10)-(head%/xscale%,y1-20),0
2520 line (tail%/xscale%,y2-10)-(tail%/xscale%,y2-20),0
2530 head%=nhead%:tail%=ntail%
2540 goto 2360
2600 rem integrate
2610 locate 5,40:print "Integrating; please wait                "
2620 sum=0:sum1=0:offs=0
2622
slope=(massint%(base2%,msptr%)-massint%(base1%,msptr%))/(base
2%-base1%)
2624 yx0=massint%(base2%,msptr%)-slope*base2%
2630 offs=((tail%+head%)/2*slope+yx0)*(tail%-head%+1)
2640 for ji=head% to tail%
2650 sum=sum+cdbl(massint%(ji,msptr%))
2652 sum1=sum1+cdbl(massint%(ji,refptr%))
2660 next ji
2670 sum=sum-abs(offs)
2680 locate 6,40:print "Mass 4 average
";int(abs(sum1/(tail%-head%+1)));
```

```
2682 locate 5,40:print "Integral= ";int(sum);"          ";
2690 goto 2400
2800 rem clear base line
2810  y1=190-(massint%(base1%,msptr%)-yoffset%)/yscale%
2820  y2=190-(massint%(base2%,msptr%)-yoffset%)/yscale%
2830 line (base1%/xscale%,y1)-(base2%/xscale%,y2),0:rem line base
line
2840  return
2980 cls:goto 1200
2990 end
3000 rem ymax% subroutine
3020 locate 1,1:print flname$;" MASS"massdisp%;"
YMAX";ymax%;"at";xmax%;
3022 print " YMIN";ymin%;"at";xmin%
3030 print "use 'ESC' key to abort display";
3031 if plotyes then print "; Plotting is selected";
3032 locate 24,1:print i;:locate 25,73:print numpoint%;
3040  return
4200  rem  output  to  plotter  -----------------
4210 if xplot%<0 then xplot%=0
4220 if xplot%>4095 then xplot%=4095
4230 if yplot%<0 then yplot%=0
4240 if yplot%>4095 then yplot%=4095
4250 plot0%=255-yplot% and 255
4260 plot2%=255-xplot% and 255
4270  plot1%=255-(int(xplot%/256)*16+int(yplot%/256))
4280 out plotter+2,plot2%:out plotter+1,plot1%:out plotter,plot0%
4290   return:rem  ----------------------------
4400  rem  saving  x,y  pair--------------------
4405 print "Make sure you have enough disk space in drive B:"
4410 input "Please enter new filename for X,Y pair ";flname1$
4412 open "b:"+flname1$ for output as #2
4420 for i=0 to numpoint%-1
4430  print#2,  i/2,massint%(i,msptr%)
4440 next i
4450 close 2
4460  return
```

TSNEWGC.TXT

```
1 REM Turbo Basic program to run GCMS for Kamala ver 3.10
04-29-90
2 on error goto 9000
4 open "mspkpar.txt" for input as 2
5
input#2,mspkstp%:input#2,mspkre%:input#2,mspkht%:input#2,msdel
ay%
6 close 2
10 KEY OFF:CLS:REM turn off key line
12 PRINT "Switch valve interface to manual"
14 PRINT "Type any key to continue"
16 IF INKEY$="" THEN 16
20 PRINT "Place DATA disk into drive B:"
40 INPUT "Enter file name you want to save on ";FLNAME$
50  mkdir
"b:\"+flname$:flnamed$=flname$:flname$="b:\"+flname$+"\"+flname$:
drive%=1
60 OPEN FLNAME$+".cmm" FOR OUTPUT AS #2
70 PRINT "Enter comments, terminate with a blank line"
80 INPUT COMMENT$:IF COMMENT$="" THEN 92
90 PRINT#2,COMMENT$:GOTO 80
92 CLOSE 2
100  BASE1=&H300
110 HEXTABLE$="0123456789ABCDEF"
120 OUT BASE1+3,&H90:REM init Metrabyte PIO12
130 INJECTOR=BASE1+1:LOOPVALVE=BASE1+2:STATES=BASE1
140 LD=2:INJECT=4:FIDSTART=1:INJNORM=&H6
150 STP=&H80:HOME=&H40:SELPOS=&H20:LOOPNORM=&HFF
160 OUT INJECTOR,INJNORM
170 OUT LOOPVALVE,LOOPNORM
180  OPEN "com1:1200,n,7,1,rs,cs,ds,cd"AS 1:REM open channel of
Omega
200  BASE2=&HE6A0
210
ATOD=BASE2:DTOA=BASE2+4:COUNTER=BASE2+8:PLOTTER=BASE2+12
220
```

```
ADSTART=BASE2+&H1C:CTRESET=BASE2+&H1D:RDYCLR=BASE2+&H1E:
SCAN=BASE2+&H1F
230 REM program 8255s of the GCMS interface
240 OUT ATOD+3,&H92:OUT DTOA+3,&H89:OUT COUNTER+3,&H92:OUT
PLOTTER+3,&H80
250 REM define dafault I/O condition
260 OUT COUNTER+2,16+4+1:REM select time-base 2 Hz
270 OUT DTOA+1,0:OUT DTOA,0:REM set mass program to 0
280 OUT PLOTTER+2,255:OUT PLOTTER+1,255:OUT PLOTTER,255:REM
zero  plotter
290 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
300 DIM MASSINT%(103,200),GC%(60,80)
310 DIM GCPKST%(1,80),gcpara%(585)
320 DIM MASSNUM%(7),MASSLOC%(1,7),LOADTIME%(15)
340 DIM GC$(2)
350 GC$(1)="FID":GC$(2)="TCD"
370 DEF FNFREESPACE(DRIVE%) 'find free space on drive
380 REG 4,DRIVE%            '0=default,1=A:,2=B:,etc.
390 REG 1,&H3600            'AH=function number
400 CALL INTERRUPT &H21     'DOS function call
410 FNFREESPACE=CSNG(REG(2))*REG(3)*REG(1)
420 END DEF
490 REM
------------------------------------------------------------
------------
500 INPUT "Please choose which GC 1. FID, 2. TCD ";GCTYPE%
502 IF (GCTYPE%<1) OR (GCTYPE%>2) THEN PRINT "WRONG TYPE":GOTO
500
510 GOSUB 5000:REM get MS parameters
520 GOSUB 5400:REM get loop loading parameters
530 INPUT "Select GC recording loop (0 to select all loops)
";GCRECORD%
540 IF (GCRECORD%<0) OR (GCRECORD%>NUMLOOP%) THEN PRINT "OUT
OF RANGE":GOTO 530
560 INPUT "Enter time (sec.) for each loop "; DUETIME%
562 PRINT
570 PRINT "Turn damping control of UTI to min."
580 PRINT "Select gain control of UTI to (-11)"
```

```
590 PRINT "Program the Omega for ramp and soak cycle"
592 PRINT "Put Omega to remote control"
594 PRINT "Switch valve interface to computer"
596 print "Turn gas valve for loading"
600 PRINT "Type any key when ready"
610 IF INKEY$="" THEN 610
620 REM DATA ACQ. START HERE-------------------------
622 MASSPG=massloc%(0,0):GOSUB 1000:REM move mass program to
1st mass
624 OUT ATOD+2,0+MASSloc%(1,0)*32:REM select 1st mass gain and
ch. 0
630 STARTDATE$=DATE$:STARTTIME$=TIME$:IF STARTDATE$<>DATE$
THEN 630
632 GOSUB 4000:REM save parameters
640 GOSUB 4600:REM do gas loop loading
642 GOSUB 1110:REM home loop valves
644 SCREEN 2:CLS:REM graphics screen
646 Print:print "Turn manual gas valve for inject"
647 print "Type any key to continue"
648 if inkey$="" then 648
649 gcpkptr%=80:mspkptr%=200:rem init. for storage need
650 FOR II=0 TO NUMLOOP%-1 :REM do gc loop one at time
654 LOCATE 20,1:PRINT "Waiting for GC loop ";II+1
656 PRINT "Last loop data:"
657 gosub 1600:rem check storage and wait if not enough
658 GCPKPTR%=0:REM reset GC peak count
660 GOSUB 2000:REM wait until gc is ready
662 open flname$+".cmm" for append as #2
664  print#2,time$,;
666 close 2
668 LOOPDATE$=DATE$:LOOPTIME=TIMER:IF LOOPDATE$<>DATE$ THEN
668
670 GOSUB 2400:REM start gc temp ramp
671 GOSUB 1360:REM switch injector valve to inject
673 for ji=0 to 500:next ji:rem delay for valve to switch
674 if II=0 then 676:rem if 1st loop skip next step
675 GOSUB 1150:REM step to next loop
676 GOSUB 2900:REM renew screen
```

```
680 GOSUB 3000:REM data acquisition procedure
682 open flname$+".cmm" for append as #2
684  print#2,time$,gcstop$
686 close 2
690 GOSUB 4200:REM data saving procedure
700 NEXT II:REM do next gc loop
702 SCREEN 0:REM text screen again
710 FINISHDATE$=DATE$:FINISHTIME$=TIME$:IF FINISHDATE$<>DATE$
THEN 710
720 PRINT "experiment starts at ";STARTDATE$;" ";STARTTIME$
730 PRINT "          finishes at ";FINISHDATE$;" ";FINISHTIME$

980 out atod+2,6*32:rem switch UTI gain back to 6
990 END:REM end of main
program-----------------------------------------------
1000  REM  output  dtoa  data----------------------------
1010 IF MASSPG<0 THEN MASSPG=0
1020 IF MASSPG>65535! THEN MASSPG=65535!
1030  MVAL=INT(MASSPG/256):LVAL=MASSPG-MVAL*256
1040 OUT DTOA+1,MVAL:OUT DTOA,LVAL
1042 if masspg>32767 then massprog%=32767-65536 else
massprog%=masspg
1050 RETURN
1100  REM  16  loop  valves  control-----------------------
1110  REM home loop valves------
1120 OUT LOOPVALVE,LOOPNORM XOR HOME
1122 FOR JI=0 TO 300:NEXT JI
1130 OUT LOOPVALVE,LOOPNORM
1132 FOR JI=0 TO 2000:NEXT JI
1140 RETURN
1150  REM  step  loop  valve-------
1160 OUT LOOPVALVE,LOOPNORM XOR STP
1162 FOR JI=0 TO 300:NEXT JI
1170 OUT LOOPVALVE,LOOPNORM
1180 RETURN
1200  REM  position  loop  valve------
1210 POX=INT(LOOPPOS/10)
1220 POX=POX*16+INT(LOOPPOS-POX*10+.2)
```

```
1230 OUT LOOPVALVE,LOOPNORM XOR POX
1240 OUT LOOPVALVE,LOOPNORM XOR (POX OR SELPOS)
1250 OUT LOOPVALVE,LOOPNORM XOR POX
1260 OUT LOOPVALVE,LOOPNORM
1270    RETURN:REM----------------------------------------
1300  REM  injector  valve  control----------------------
1310  REM  load  injector-----------
1320 OUT INJECTOR,INJNORM XOR LD
1330 FOR JI=0 TO 1500:NEXT JI:REM delay
1340 OUT INJECTOR,INJNORM
1350 RETURN
1360   REM   inject------------------
1370 OUT INJECTOR,INJNORM XOR INJECT
1380 FOR JI=0 TO 1500:NEXT JI:REM delay
1390 OUT INJECTOR,INJNORM
1400 RETURN
1410    REM    ----------------------------------------------
1500  REM  wait  fid  ready------------------------------
1502 FOR JI=0 TO 20
1510 IF INP(STATES) AND 1 THEN JI=0:REM test until FID is ready
1512 NEXT JI
1520 RETURN
1530  REM  FID  remote  start--------
1540 OUT INJECTOR,INJNORM XOR FIDSTART
1550 FOR JI=0 TO 300:NEXT JI:REM delay
1560 OUT INJECTOR,INJNORM
1570 RETURN
1580     REM----------------------------------------------
1600 rem check drive B: storage and wait if not enough--
1610 mspkneed%=mspkptr%+10:if mspkneed%>200 then
mspkneed%=200
1620 IF (GCRECORD%=0) OR (GCRECORD%=II+1) then
gcpkneed%=gcpkptr%+5 else gcpkneed%=0
1622 if gcpkneed%>80 then gcpkneed%=80
1630  need1=104*cdbl(mspkneed%)*2
1640  need2=61*cdbl(gcpkneed%)*2
1650 need=need1+need2
1660 if need < fnfreespace(2) then 1690
```

```
1670 print "Not enough disk space; put another formated disk in B:"
1680 print "Type 'return' key to continue"
1682 if inkey$<> chr$(13) then 1682
1684 if need >= fnfreespace(2) then 1670
1686 mkdir "b:\"+flnamed$
1690    return:rem----------------------------------------
2000 REM wait for gc to be ready ---------------------
2010 PRINT "CHECKING " GC$(GCTYPE%)
2020 ON GCTYPE% GOSUB 1500,2100:REM fid or tcd
2030 PRINT "GC IS READY"
2040 RETURN:REM appropiate gc is ready----------------
2100 REM wait until TCD is ready--------
2110 GOSUB 2200:REM read Omega PSW
2120 IF (OMEPSW AND 128) THEN 2110:REM test Omega stop flag
2130 RETURN:REM TCD is ready -----------
2200 REM read Omega PSW ----------------
2210 PRINT#1,"*00:RPSW:":REM command Omega to read
2220 INPUT#1,OME$:IF MID$(OME$,2,2)<>"00" THEN PRINT
"error";OME$:STOP
2230 PSW$=MID$(OME$,5,2)
2240
OMEPSW=(INSTR(LEFT$(PSW$,1),HEXTABLE$)-1)*16+INSTR(RIGHT$(P
SW$,1),HEXTABLE$)-1
2250   RETURN:REM-----------------------
2400 REM start GC temp ramp--------------------------
2410 PRINT:PRINT "STARTING ";GC$(GCTYPE%)
2420 ON GCTYPE% GOSUB 1530,2500:REM fid or tcd start
2430 PRINT GC$(GCTYPE%);" IS STARTED"
2440 PRINT
2450    RETURN:REM-------------------------------------
2500   REM   START   TCD-----------------------------------
2510 GOSUB 2200:REM read Omega PSW
2520 NEWPSW=PSW OR 128:REM set start bit for Omega
2530 PRINT #1,"00:wpsw/";HEX$(NEWPSW);":"
2540 INPUT#1,OME$:REM get reponse
2550 IF MID$(OME$,2,2)<>"00" THEN PRINT "error";OME$:STOP
2560    RETURN:REM-------------------------------------
2600 REM check if gc is running return with gcstop% --------
```

```
2610 ON GCTYPE% GOSUB 2700,2800:REM fid,tcd stop ?
2612 if gcstop%=1 then sense%=sense%+1 else sense%=0
2614 if sense%<9 then gcstop%=0 else gcstop$="Stop by GC"
2620 LASPDATE$=DATE$:LASPTIME=TIMER:IF LASPDATE$<>DATE$
THEN 2620
2630 IF LASPDATE$<>LOOPDATE$ THEN LASPTIME=LASPTIME+86400!
2640 IF (LASPTIME-LOOPTIME)>DUETIME% THEN
GCSTOP%=1:gcstop$="Timer Stop"
2690   RETURN:REM--------------
2700 REM check if fid stop
2710 IF INP(STATES) AND 1 THEN GCSTOP%=0 ELSE GCSTOP%=1
2720   RETURN:REM--------------
2800 REM check if tcd stop
2810 GOSUB 2200:REM read Omega PSW
2820 IF (OMEPSW AND 128) THEN GCSTOP%=0 ELSE GCSTOP%=1
2830
RETURN:REM-----------------------------------------------
2900  REM  init  screen---------
2910 CLS:SCRNX%=0
2912 LOCATE 20,1:PRINT "GC loop ";II+1
2920 LOCATE 21,1
2930 PRINT "GC peak found = ";GCPKPTR%
2940 print "MS peak found = ";mspkptr%
2990   RETURN:REM--------------
3000 REM data acquisition
procedure---------------------------------
3002 PASSFLAG=0:OLDCOUNT=0:ELEMENT%=0:GCPKPTR%=0:REM init.
counts & ptrs
3003 LASTCOUNT%=0:KNEE%=1:TAIL%=1:GCPKST%=0
3004 INDPTR%=0:GCSTOP%=0:SCRNX%=0:REM init. massbuf% index and
stop  flag
3006 ERASE MASSINT%,GC%,GCPKST%,gcpara%
3007
gcpara%(6)=mspkstp%:gcpara%(7)=mspkre%:gcpara%(9)=mspkht%
3008  gcpara%(8)=ms4%:gcpara%(20)=100:gcpara%(26)=msdelay%
3009  gcpara%(2)=gctype%:gcpara%(5)=element:gcpara%(12)=gcpkptr%
3010 IF (INP(ATOD+1) AND &H20) =0 THEN 3010:REM WAIT UTIL
COUNTER IS READY
```

```
3012 OUT RDYCLR,0:REM clear counter ready flag
3016 if element%< (mslag%*2) then gcpara%(19)=0 else
gcpara%(19)=nummass%

3020 call readgc
(massint%(0,0),massloc%(0,0),gc%(0,0),gcpkst%(0,0),gcpara%(0))
3060 gccount%=gcpara%(3):ms%=gcpara%(156+(dispmass%-1)*61)
3062 if mspkptr% <> gcpara%(24) then mspkptr%=gcpara%(24):gosub
2920
3070 if gcpkptr% <> gcpara%(12) then gcpkptr%=gcpara%(12):gosub
2920
3075 IF SCRNX%>639 THEN GOSUB 2900:REM clear screen if full
3076 if element%=0 then 3190 :rem no display
3077 print ms%,gcpara%(27);

3080 gcscrny%=199-(gccount%-gcpara%(6))/20
3082 if gcscrny% <100 then gcscrny%=100
3084 msscrny%=99-ms%/20
3086 if msscrny% < 0 then msscrny%=0
3100 IF SCRNX%=0 THEN LINE
(0,gcscrny%)-(0,gcscrny%):oldgcy%=gcscrny%
3102 if scrnx%=0 then line
(0,msscrny%)-(0,msscrny%):oldmsy%=msscrny%:goto 3180
3104 line (scrnx%-1,oldgcy%)-(scrnx%,gcscrny%):oldgcy%=gcscrny%
3106 line
(scrnx%-1,oldmsy%)-(scrnx%,msscrny%):oldmsy%=msscrny%
3180 SCRNX%=SCRNX%+1:REM prepare for next dot

3190 REM end of gc peak----------
3200 REM read masses----------


3410 ELEMENT%=ELEMENT%+1
3412 LOCATE 23,1:PRINT "At point ";ELEMENT%
3420 IF GCSTOP%=0 THEN REFDATE$=DATE$:REFTIME=TIMER
3430 IF GCSTOP%=0 THEN GOSUB 2600:REM check if gc is still
running
3450 NOWTIME=TIMER:IF REFDATE$<>DATE$ THEN
```

```
NOWTIME=NOWTIME+86400!
3460 IF NOWTIME-REFTIME<240 THEN 3010:REM continue data acq.
for 4 min.
3470 RETURN:REM data acq proedure
return-----------------------------
4000 REM Saving parameters routine--------------------------
4010 PRINT "Saving parameters; please wait"
4020 OPEN FLNAME$+".par" FOR OUTPUT AS #2
4030 PRINT#2,GC$(GCTYPE%)
4040 PRINT#2,NUMMASS%
4050 FOR I=0 TO NUMMASS%-1
4060 PRINT#2,MASSnum%(I),MASSloc%(1,I) :rem save massnum and
gain
4070 NEXT I
4080 PRINT#2,NUMLOOP%
4090 FOR I=0 TO NUMLOOP%-1
4100 PRINT#2,LOADTIME%(I)
4110 NEXT I
4120 PRINT#2,GCRECORD%
4130 PRINT#2,DUETIME%
4140 CLOSE 2:RETURN:REM ----------------
4200 REM saving data ------------------
4202 LOCATE 1,1
4204 PRINT "Saving data, please wait"
4206 IF (GCRECORD%=0) OR (GCRECORD%=II+1) THEN 4210 ELSE 4290
4210 OPEN FLNAME$+".a"+MID$(STR$(II),2) FOR OUTPUT AS #2
4212 gcpkptr%=gcpara%(12):print#2,gcpkptr%
4220 for i=0 to gcpkptr%-1
4230 print#2,gcpkst%(0,i),gcpkst%(1,i):rem save peak elem & count
4240 next i
4250 CLOSE 2
4252 if gcpkptr%=0 then 4290
4260 DEF SEG=VARSEG(GC%(0,0)):REM get seg for BASVE
4270 LENGTH=VARPTR(GC%(60,GCPKPTR%-1))+2:REM get length for
BASVE
4280 BSAVE FLNAME$+".g"+MID$(STR$(II),2),0,LENGTH
4290 if mspkptr%=0 then 4590
4310 OPEN FLNAME$+".b"+MID$(STR$(II),2) FOR OUTPUT AS #2
```

```
4320 print#2,mspkptr%:rem save # of mass peaks
4330 CLOSE 2
4340 DEF SEG=VARSEG(MASSINT%(0,0))
4350 LENGTH=VARPTR(MASSINT%(103,mspkptr%-1))+2
4360 BSAVE FLNAME$+".m"+MID$(STR$(II),2),0,LENGTH
4590 RETURN:REM=========
4600 REM GC gas loading
loop-----------------------------------------
4602 PRINT:PRINT "LOADING LOOP VALVES"
4610 GOSUB 1100:REM home loop valves
4620 GOSUB 1310:REM put injector valve to load
4622 if numloop%=1 then 4700:rem 1st loop only
4630 FOR I=1 TO NUMLOOP%-1:REM do every loop
4632 GOSUB 1150:REM step loop valve to next loop
4633 if i <> 2 then 4636  else beep:print "Type any key to continue
LOOP 3"
4634 if inkey$="" then 4634
4636 PRINT "LOADING LOOP";I+1;" FOR";LOADTIME%(I);" SEC."
4640 REFDATE$=DATE$:REFTIME=TIMER:IF REFDATE$<>DATE$ THEN
4640:REM get ref time
4650 IF REFDATE$<>DATE$ THEN DAY=1 ELSE DAY=0
4660 PASSTIME=TIMER+DAY*86400!:REM account for day boundary
4670 IF (PASSTIME-REFTIME)<LOADTIME%(I) THEN 4650:REM wait for
full load time
4690 NEXT I:REM fill all gas loops
4700 print "LOADING LOOP 1 FOR";LOADTIME%(0);" SEC."
4702 gosub 1100:rem home loop valve
4710 REFDATE$=DATE$:REFTIME=TIMER:IF REFDATE$<>DATE$ THEN
4710
4720 IF REFDATE$<>DATE$ THEN DAY=1 ELSE DAY=0
4730 PASSTIME=TIMER+DAY*86400
4740 IF (PASSTIME-REFTIME)<LOADTIME%(0) THEN 4720
4800
PRINT:RETURN:REM-----------------------------------------
-----
5000 REM get MS parameters ---------------------------
5010 INPUT "Enter number of masses interested ";NUMMASS%
5012 PRINT
```

```
5014 if nummass%=0 then 5300
5020 PRINT "Locate mass :"
5030 PRINT "     to the prompt of 'LOCATE' use keypad key"
5040 PRINT "       '8' for fine up"
5050 PRINT "       '2' for fine down"
5060 PRINT "       '6' for coarse up"
5070 PRINT "       '4' for coarse down"
5080 PRINT "     to move mass around "
5082 PRINT "         'K'     to use key entry
5084 print "         'G'     to select gain"
5090 PRINT "     and 'ENTER' to select location":PRINT
5100 MASSPG=0:GOSUB 1000:REM preset to mass 0
5110 FOR I=0 TO NUMMASS%-1
5120 PRINT I+1;"  ";
5130 INPUT "enter mass number ";MASSNUM%(I)
5132 if massnum%(i)=4 then ms4%=i+1
5140 PRINT "LOCATE    ";
5150 TEMP$=INKEY$:IF TEMP$="" THEN 5150
5160 IF ASC(TEMP$)=13 THEN 5250:REM select location of mass
5162 if temp$="K" then 5242
5164 if temp$="G" then gosub 5500:rem change gain%
5170 IF LEN(TEMP$)<>2 THEN 5150
5180 TEMP=ASC(RIGHT$(TEMP$,1))
5190 IF TEMP=72 THEN MASSPG=MASSPG+1
5200 IF TEMP=80 THEN MASSPG=MASSPG-1
5210 IF TEMP=77 THEN MASSPG=MASSPG+100
5220 IF TEMP=75 THEN MASSPG=MASSPG-100
5230 GOSUB 1000:REM program D to A
5240 GOTO 5150
5242 input "Enter D to A setting ";MASSPG
5244  goto  5230
5250 PRINT "at D to A setting of ";MASSPG
5260 MASSLOC%(0,I)=MASSProG%
5270  MASSloc%(1,I)=gain%
5272  print "gain control for mass";massnum%(i);" is";gain%
5280 PRINT
5290 NEXT I
5291 if ms4%=0 then beep:print "Mass 4 not selected" else
```

```
ms4%=ms4%-1
5292 print "Display which mass (1 to";nummass%;") ";
5294 input dispmass%
5296 input "MS delay after GC (sec.) ";mslag%
5300
RETURN:REM-----------------------------------------------
5400 REM get loop valve load data ----------------------------
5410 PRINT "GETTING LOOP VALVE LOAD DATA"
5412 print "LOOP 1 WILL BE YOUR 1ST LOOP WITH STEADY LOADING"
5414 PRINT "Load-Time for loop 1 is the min. load-time for it"
5420 INPUT "Enter number of loops (1 to 16)"; NUMLOOP%
5430 IF (NUMLOOP%<1) OR (NUMLOOP%>16) THEN 5420
5440 FOR I=0 TO NUMLOOP%-1
5450 PRINT "enter load-time (in sec.) for loop";I+1;
5460 INPUT LOADTIME%(I)
5470 NEXT I
5480 RETURN:REM
-----------------------------------------------
5500 rem change gain%---------------
5510 input "enter mass gain (0 to 7) ";gain%
5520 if (gain%<0) or (gain%>7) then 5510:rem out of range
5550 out atod+2,gain%*32:rem switch gain
5560   return:rem--------------------
9000 rem error handle routine-------------------------------
9010 open "tberr1.txt" for output as 10
9020 print#10,erl,err
9030 close 10
9032 print "ERR";err;"AT line";erl
9040
end:rem-----------------------------------------------------

10000 SUB READGC INLINE
10010 $inline "gcms.bin"
10020 end sub
```

READTS.TXT

```
100 REM loading data from TSNEWGC program ver 3.10 4-27-90 by
Henry Chan
110 on error goto 9000
120  base2=&he6a0:plotter=base2+12
130  out  plotter+3,&h80
140  xplot%=0:yplot%=0:gosub  6400:rem  zero  plotter
200 DIM GCELEM%(80)
210 DIM
MASSPEAK%(7),GCPEAK%(80),pkbg%(200),pkend%(200),pkbase%(200)
220 DIM MASSNUM%(7),MASSGAIN%(7),LOADTIME%(15)
230  dim  gcret%(100),msdel%(100)
232  cls:print "Loading  reference  data"
240  open "readts1.par" for input as 1
250  for i=0 to 93
260   input#1,gcret%(i),msdel%(i)
270  next i
280  close 1
310  PRINT "Place data disk in drive B:, type any key to continue"
312  IF INKEY$="" THEN 312
314  chdir  "b:\"
316  files  "b:*."
320  INPUT "Enter file name ";FLNAME$
322  chdir  "b:"+flname$
330  OPEN "b:"+FLNAME$+".cmm" FOR INPUT AS #2
340  WHILE NOT EOF(2)
350  LINE INPUT#2,COMMENT$
360  PRINT COMMENT$
370  WEND
380  CLOSE 2
390  INPUT "Is this the right one (y/n) ";YES$
400  IF YES$="y" THEN 430
410  IF YES$="n" THEN 310
420  GOTO 390
430  PRINT
500  REM Loading
parameters----------------------------------------
```

```
510 PRINT "Loading parameters, please wait"
520 OPEN "b:"+FLNAME$+".par" FOR INPUT AS #2
530 INPUT#2,GC$:PRINT "GC type was ";GC$
540 INPUT#2,NUMMASS%:PRINT "Number of masses observed
=";NUMMASS%
550 FOR I=0 TO NUMMASS%-1
560 INPUT#2, MASSNUM%(I),MASSGAIN%(I)
570 PRINT "MASS";I+1;"is";MASSNUM%(I),"Gain factor";MASSGAIN%(I)
580 NEXT I
590 INPUT#2,NUMLOOP%:PRINT "Number of GC loop is "NUMLOOP%
600 FOR I=0 TO NUMLOOP%-1
610 INPUT#2,LOADTIME%(I)
620 PRINT "Loop";I+1;" load time is ";LOADTIME%(I);" sec."
630 NEXT I
640 INPUT#2,GCRECORD%:PRINT "GC recording loop is (0 for
all)";GCRECORD%
650 INPUT#2,DUETIME%:PRINT "Max. time for GC loop is";DUETIME%;"
sec."
660 CLOSE 2:REM parametes
loaded----------------------------------
700 REM display
parameters--------------------------------------
710 CLS
720 GOSUB 1700:REM display parameters----------
1000 REM read
data-------------------------------------------------
1010 PRINT:
1020 INPUT "Enter which GC loop (0 to exit) ";II
1022 if ii=0 then 9900:rem exit program
1030 IF (II<1) OR (II>NUMLOOP%) THEN PRINT "Out of range":GOTO
1020
1050 IF (GCRECORD%=0) OR (GCRECORD%=II) THEN 1060 ELSE GOTO
1200
1060 REM load gc data-----
1070 OPEN "b:"+FLNAME$+".a"+MID$(STR$(II-1),2) FOR INPUT AS#2
1080 INPUT#2,GCPKPTR%:PRINT "number of GC peak located
is";GCPKPTR%
1082 if gcpkptr%=0 then close 2:dim gc%(60,gcpkptr%+1):goto 1200
```

```
1090 FOR I=0 TO gcpkptr%-1
1100 INPUT#2,GCELEM%(I),GCPEAK%(I)
1110 NEXT I
1112 CLOSE 2
1150 PRINT "Loading GC data, please wait"
1152  gcname$="b:"+flname$+".g"+mid$(str$(ii-1),2)
1158 dim gc%(60,gcpkptr%+1)
1160 DEF SEG=VARSEG(GC%(0,0))
1170 BLOAD gcname$
1180 PRINT "Done loading GC data"
1200  REM  read  MS  data----------
1202 if nummass%=0 then dim massint%(103,mspkptr%+1):goto
1400
1210 OPEN "b:"+FLNAME$+".b"+MID$(STR$(II-1),2) FOR INPUT AS #2
1230 INPUT#2,mspkptr%: rem get # of mass peak%
1260 CLOSE 2
1262 PRINT
1264  dim  massint%(103,mspkptr%+1)
1270 DEF SEG=VARSEG(MASSINT%(0,0))
1280 PRINT "Loading MS data, please wait"
1290 BLOAD "b:"+FLNAME$+".m"+MID$(STR$(II-1),2)
1300 PRINT "Done loading MS data"
1400    REM    Data    view----------------------------------------
1402 CLS:GOSUB 1700:GOSUB 1900:REM print peaks data
1410 PRINT "To view 1. GC peak, 2. MS peak, 3. another loop, 4. Auto
align"
1420 INPUT "Enter choice by number ";CHOICE
1430 ON CHOICE GOTO 1450,1460,1470,1462
1440 GOTO 1410:REM no choice
1450 GOSUB 2000:GOTO 1400 :REM view GC peak
1460 GOSUB 3000:GOTO 1400 :REM view MS peak
1462 gosub 7000:goto 1400 :rem auto align GC & MS data
1470 erase gc%:erase massint%:goto 700:rem clear gc% array and
repeat
1480 gosub 4000:goto 1400:rem auto-integrate GC peaks
1700 REM print parameters
1720 PRINT "GC type used was ";GC$
1730 PRINT "Number of masses observed ="NUMMASS%
```

```
1740 FOR I=0 TO NUMMASS%-1
1750 PRINT "mass";MASSNUM%(I)" gain";MASSGAIN%(I),;
1760 NEXT I
1770 PRINT
1780 PRINT "Number of GC loop is";NUMLOOP%
1790 PRINT "Load time in sec. for each loop"
1800 FOR I=0 TO NUMLOOP%-1
1810 PRINT I+1;" is";LOADTIME%(I),;
1820 NEXT I
1830 PRINT
1840 PRINT "GC recording loop (0 for all) is";GCRECORD%
1850 PRINT "Max. time for each GC loop is";DUETIME%;"sec."
1860    RETURN:REM-----------------------------------
1900 REM print GC & MS data--------
1910 PRINT "LOOP #";ii;" GC peaks located =";GCPKPTR%
1912 print "GC peak#, at element ,intensity"
1920 FOR I=0 TO GCPKPTR%-1
1930 PRINT I+1;GCELEM%(I)/2;GCPEAK%(I);
1932  tabcount=(i+1)-int((i+1)/4+.1)*4:print  tab(tabcount*20);
1940 NEXT I
1950 print:PRINT "Total mass peak ";mspkptr%
1990  RETURN:REM  ------------------
2000 REM view GC peak
2002 GCMAX=0:if gcpkptr%=0 then print "no GC peak":return
2003 FOR I=0 TO GCPKPTR%-1
2004 IF GCPEAK%(I) > GCMAX THEN GCMAX=GCPEAK%(I)
2005  next i:print:gosub  1900
2006 INPUT "Which gc peak (e to exit,p to plot,a to
autointegrate)";GCPOINT$
2007  gcpoint=val(gcpoint$):if  gcpoint$="e"   then  return
2008 if gcpoint$="a" then gosub 4000:goto 2000
2009 if gcpoint$="p" then gosub 6000:goto 2000
2010 if  ((gcpoint=0) or (gcpoint>gcpkptr%)) then 2000
2011 SCREEN 2:YSCALE=GCMAX/180:gosub 4500
2012  nhead=head:ntail=tail
2013 cls:line (30,0)-(30,10):rem tack peak
2014 SCRNY=199-(GC%(0,GCPOINT-1)-baseline+10)/YSCALE
2016 IF SCRNY < 0 THEN SCRNY=0
```

```
2020 LINE (0,SCRNY)-(0,SCRNY):rem 1st point
2021 locate 1,40:print "Loop #";ii
2022 locate 2,40:print "GC peak";gcpoint;"
at";gcelem%(gcpoint-1)/2;"sec."
2024 locate 3,40:print gcpeak%(gcpoint-1)
2030 FOR I=1 TO 60
2032 SCRNY=199-(GC%(I,GCPOINT-1)-baseline+10)/YSCALE
2034 IF SCRNY < 0 THEN SCRNY=0
2040 LINE -(I,SCRNY)
2050 NEXT I
2200  y1=199-(baseline-baseline+10)/yscale
2216  line  (0,y1)-(60,y1)
2222  y1=199-(gc%(head,gcpoint-1)-baseline+10)/yscale
2224  y2=199-(gc%(tail,gcpoint-1)-baseline+10)/yscale
2226  line  (head,y1)-(head,y1-5)
2228  line  (tail,y2)-(tail,y2-5)
2230 ky$=inkey$:if ky$="" then 2230
2240 if asc(ky$)=13 then 2500:rem CR
2250 if asc(ky$)=27 then 2950:rem ESC
2251 if ky$="R" then goto 2013
2252 if ky$="E" then yscale=yscale/2:goto 2013
2253 if ky$="S" then yscale=yscale*2:goto 2013
2254 if ky$="<" then baseline=baseline-1:goto 2013
2255 if ky$=">" then baseline=baseline+1:goto 2013
2258 if ky$="^" then gosub 2700:goto 2230:rem display array
2260 if len(ky$)<2 then 2230
2262 locate 5,40
2264 print "                    ";
2270  temp=asc(right$(ky$,1))
2280 if temp=72 then if head<59 then nhead=head+1:gosub 2800
2290 if temp=80 then if head>0 then nhead=head-1:gosub 2800
2300 if temp=77 then if tail<59 then ntail=tail+1:gosub 2850
2310 if temp=75 then if tail>0 then ntail=tail-1:gosub 2850
2340 head=nhead:tail=ntail:goto 2230
2500  rem  integrate----
2510 sum=0:if head=tail then 2600
2530 for ji=head to tail
2550  sum=sum+gc%(ji,gcpoint-1)-baseline
```

```
2560 next ji
2600 locate 5,40
2610 print "INTEGRAL =";sum;"      ";
2620 goto 2200
2700 rem display array--------
2710 for i=0 to 60
2715 if i=30 then print "*";
2720 print gc%(i,gcpoint-1),;
2730 next i
2740 print
2750   return:rem---------------
2800 rem tack header----------
2805 if nhead>ntail  then nhead=ntail
2810 line(head,y1-1)-(head,y1-5),0:rem erase old tack
2815 y1=199-(gc%(nhead,gcpoint-1)-baseline+10)/yscale:rem
calculate new y1
2820 line(nhead,y1)-(nhead,y1-5):rem tack new leader
2830   return:rem---------------
2850 rem tack tail
2855 if ntail<nhead then ntail=nhead
2860 line(tail,y2-1)-(tail,y2-5),0:rem erase old tack
2865 y2=199-(gc%(ntail,gcpoint-1)-baseline+10)/yscale:rem
calculate new y2
2870 line(ntail,y2)-(ntail,y2-5):rem tack new tail
2880   return:rem----------------
2950 LOCATE 22,1
2960 PRINT "Type any key to exit"
2970 IF INKEY$="" THEN 2970
2980 SCREEN 0
2990 goto 2000:rem --------------
3000 rem view MS data----------
3002 if mspkptr%=0 then print "no mass peak":return
3006 print "View mass peaks"
3010 for i=0 to nummass%-1
3020 print "Mass";massnum%(i);" gain";massgain%(i),;
3030 next i
3040 print
3070 input "Which Mass (e to exit)";mass$
```

```
3072 mass=val(mass$):if mass$="e" then return
3080 mspoint=0
3090 for i=0 to nummass%-1
3100 if mass=massnum%(i) then mspoint=i+1
3110 next i
3120 if mspoint=0 then print "Mass not found":goto 3070
3122 mpmax%=0:mspk%=0:for i=0 to mspkptr%-1
3123 if massint%(0,i)<>(mspoint-1) then 3127
3124 if massint%(2,i) > mpmax% then mpmax%=massint%(2,i)
3126 mspk%=mspk%+1:print mspk%;massint%(1,i)/2;massint%(2,i),;
3127 next i:print
3128 print mspk%;"peaks for mass";massnum%(mspoint-1)
3129 input "Which one (e to exit,p to plot, a to autointegrate)";mk$
3130 if mk$="e" then 3000
3131 if mk$="a" then gosub 5000:goto 3120:rem autointegrate
3132 if mk$="p" then gosub 6500:goto 3120:rem plot
3133 mk%=val(mk$):if mk%>mspk% then print "out of range":goto
3129
3134 dim vmloc%(mspk%):mspk%=0:for i=0 to mspkptr%-1
3135 if massint%(0,i)<>(mspoint-1) then 3137
3136 vmloc%(mspk%)=massint%(1,i):mspk%=mspk%+1:if mspk%=mk%
then vmspk%=i
3137 next i
3138 yscale=mpmax%/180:gosub 4700:erase vmloc%:rem find
baseline
3139  scrny1=199-(massint%(3,vmspk%)-baseline+10)/yscale
3142  scrny2=199-(massint%(54,vmspk%)-baseline+10)/yscale
3145  nhead=head:ntail=tail
3150 screen 2:cls
3152 line (25,0)-(25,10):rem tack peak
3160 line (0,scrny1)-(0,scrny1):rem point 1st pt.
3162 locate 1,40:print "Loop #";ii
3170 locate 2,40:print "Mass";mass;" peaks
at";massint%(1,vmspk%)/2;"sec."
3190 locate 3,40:print massint%(2,vmspk%)
3200 for i=4 to 53
3202  oscrny1=scrny1
3210  scrny1=199-(massint%(i,vmspk%)-baseline+10)/yscale
```

```
3220 if scrny1 < 0 then scrny1=0
3230  line  (i-4,oscrny1)-(i-3,scrny1)
3232  oscrny2=scrny2
3233  scrny2=199-(massint%(i+50,vmspk%)-baseline+10)/yscale
3234 if scrny2 < 0 then scrny2=0
3235  line  (i+100,oscrny2)-(i+101,scrny2)
3240  next i
3300  y1=199-(baseline-baseline+10)/yscale
3320 LINE (0,Y1)-(50,Y1):REM draw base line
3330  Y1=199-(massint%(HEAD,vmspk%)-baseline+10)/YSCALE
3340  Y2=199-(massint%(TAIL,vmspk%)-baseline+10)/YSCALE
3350  LINE  (HEAD-3,Y1)-(HEAD-3,Y1-5)
3360  LINE  (TAIL-3,Y2)-(TAIL-3,Y2-5)
3370 KY$=INKEY$:IF KY$="" THEN 3370
3380 IF ASC(KY$)=13 THEN 3520:REM CR
3390 IF ASC(KY$)=27 THEN 3820:REM ESC
3400 IF KY$="R" THEN GOTO 3150
3410  if  ky$="E"  then  yscale=yscale/2:goto  3150
3412  if  ky$="S"  then  yscale=yscale*2:goto  3150
3413  if  ky$="<"  then  baseline=baseline-1:goto  3150
3414  if  ky$=">"  then  baseline=baseline+1:goto  3150
3420 IF KY$="^" THEN GOSUB 3630:GOTO 3370:REM display array
3430 IF LEN(KY$)<2 THEN 3370
3440 LOCATE 5,40
3450 PRINT "                    ";
3460 TEMP=ASC(RIGHT$(KY$,1))
3470 IF TEMP=72 THEN IF HEAD<53 THEN NHEAD=HEAD+1:GOSUB 3700
3480 IF TEMP=80 THEN IF HEAD>3 THEN NHEAD=HEAD-1:GOSUB 3700
3490 IF TEMP=77 THEN IF TAIL<53 THEN NTAIL=TAIL+1:GOSUB 3760
3500 IF TEMP=75 THEN IF TAIL>3 THEN NTAIL=TAIL-1:GOSUB 3760
3510 HEAD=NHEAD:TAIL=NTAIL:GOTO 3370
3520  REM  integrate----
3530 SUM=0:sum4=0:IF HEAD=TAIL THEN 3600
3560 FOR JI=HEAD TO TAIL
3580  SUM=SUM+massint%(JI,vmspk%)-BASELINE
3582  sum4=sum4+massint%(ji+50,vmspk%)
3590 NEXT JI
3600 LOCATE 5,40
```

```
3610 PRINT "INTEGRAL =";SUM;"     ";
3612 locate 6,40 :print "Average mass 4 intensity"
;sum4/(tail-head+1)
3620 GOTO 3300
3630  REM  display  array--------
3640 FOR I=0 TO 103
3650 IF I=28 THEN PRINT "*";
3652 if i=base1 then print "<";
3654 if i=base2 then print ">";
3660  PRINT  massint%(I,vmspk%),;
3670 NEXT I
3680 PRINT
3690   RETURN:REM---------------
3700  REM  tack  header----------
3710 IF NHEAD>NTAIL  THEN NHEAD=NTAIL
3720  LINE(HEAD-3,Y1-1)-(HEAD-3,Y1-5),0:REM erase old tack
3730  Y1=199-(massint%(NHEAD,vmspk%)-baseline+10)/YSCALE:REM
calculate  new  y1
3740 LINE(NHEAD-3,Y1)-(NHEAD-3,Y1-5):REM tack new leader
3750   RETURN:REM---------------
3760 REM tack tail
3770 IF NTAIL<NHEAD THEN NTAIL=NHEAD
3780  LINE(TAIL-3,Y2-1)-(TAIL-3,Y2-5),0:REM  erase  old  tack
3790  Y2=199-(massint%(NTAIL,vmspk%)-baseline+10)/YSCALE:REM
calculate  new  y2
3800  LINE(NTAIL-3,Y2)-(NTAIL-3,Y2-5):REM  tack  new  tail
3810   RETURN:REM---------------
3820 LOCATE 22,1
3830 PRINT "Type any key to exit"
3840 IF INKEY$="" THEN 3840
3850 SCREEN 0
3860  goto  3120:REM  -------------
4000  rem  auto-integrate
4002  print "GC auto-integrate loop #";ii
4010 dim area(gcpkptr%):rem clear array
4020 gosub 4200:rem auto routine
4030  print
4040 input "Do you want to save integrals (y/n)"; yes$
```

```
4050 if yes$="y" then gosub 4300:goto 4080
4060 if yes$="n" then 4080
4070 goto 4040
4080 input "Do you want to print integrals (y/n)"; yes$
4090 if yes$="y" then gosub 4400:goto 4120
4100 if yes$="n" then 4120
4110 goto 4080
4120   erase   area:return:rem-----------------------------
4200 for gcpk%=0 to gcpkptr%-1:rem do it for every GC peak
4210 gcpoint=gcpk%+1:gosub 4500:rem look baseline & define peak
4220 sum=0:sum4=0: if head=tail then 4290
4230 pkbase%(gcpk%)=baseline
4240 for ji=head to tail
4250 sum=sum+gc%(ji,gcpk%)-baseline
4260 next ji
4270 area(gcpk%)=fix(sum)
4280 print gcpk%+1;" ";area(gcpk%),;
4290 next gcpk%
4299    return:rem-----------------------------------------
4300 rem saving integrals
4302 sname$=left$(flname$,4)+"gc"+mid$(str$(ii),2)+".prn"
4310 open "b:"+sname$ for output as # 2
4312 print#2,"peak","at (sec.)","height","area","baseline"
4330 for gcpk%=0 to gcpkptr%-1
4340 print#2, gcpk%+1,gcelem%(gcpk%)/2,gcpeak%(gcpk%),;
4342 print#2, area(gcpk%),pkbase%(gcpk%)
4350 next gcpk%
4360 print#2,
4370 close 2
4390    return:rem-------------------------------
4400  rem  print  integral-----------------------
4410 lprint flname$;"GC peaks for  Loop #";ii
4420 lprint
4428 lprint "peak","at (sec.)","height","area","baseline"
4430 NL%=0:for gcpk%=0 to gcpkptr%-1
4440 lprint
gcpk%+1,gcelem%(gcpk%)/2,gcpeak%(gcpk%),area(gcpk%),pkbase%(gc
pk%)
```

```
4450 NL%=NL%+1:IF NL%=5 THEN NL%=0:LPRINT
4480 next gcpk%
4482  lprint:lprint
4490     return:rem--------------------------------
4500 rem find base1 & base2
4501 gp=gcpoint
4502 if gp=1 then head=0 else
head=30-(gcelem%(gp-1)-gcelem%(gp-2))
4504 if gp=gcpkptr% then tail=60 else
tail=(gcelem%(gp)-gcelem%(gp-1))+30
4506 if head < 0 then head=0
4508 if tail >60 then tail=60
4582 a%=4:dim a(a%):for i=0 to 4:a(i)=4096:next i
4583 for i=5 to 60:if gc%(i,gcpoint-1) > a(0) then 4588
4584  a(0)=gc%(i,gcpoint-1)
4585 for i1=0 to 3
4586 if a(i1) < a(i1+1) then a1=a(i1):a(i1)=a(i1+1):a(i1+1)=a1
4587 next i1
4588 next i
4589 sum=0:for i=0 to 4:sum=sum+a(i):next i:baseline=sum/5:erase a
4590 basemin%=gc%(head,gcpoint-1):nhead=head
4600 for i=head to 30
4610 if gc%(i,gcpoint-1) < basemin% then
nhead=i:basemin%=gc%(i,gcpoint-1)
4620 next i
4630  basemin%=gc%(31,gcpoint-1):ntail=31
4640 for i=31 to tail
4650 if gc%(i,gcpoint-1) <= basemin% then
ntail=i:basemin%=gc%(i,gcpoint-1)
4660 next i
4670  head=nhead:tail=ntail
4680 if head < 15 then head=15
4682 if tail > 52 then tail=52
4690     return:rem--------------------------------
4700 rem find ms baseline
4710 if mk%=1 then head=0 else
head=27-(vmloc%(mk%-1)-vmloc%(mk%-2))
4720 if mk%=mspk% then tail=53 else
```

```
tail=(vmloc%(mk%)-vmloc%(mk%-1))+27
4730 if head < 2 then head=2
4732 if tail > 53 then tail=53
4740 a%=4:dim a(a%):for i=0 to 4:a(i)=4096:next i
4745 for i=2 to 53:if massint%(i,vmspk%) > a(0) then 4770
4750 a(0)=massint%(i,vmspk%)
4755 for i1=0 to 3
4760 if a(i1) < a(i1+1) then a1=a(i1):a(i1)=a(i1+1):a(i1+1)=a1
4765 next i1
4770 next i
4775 sum=0:for i=0 to 4:sum=sum+a(i):next i:baseline=sum/5:erase a
4780 basemin%=massint%(head,vmspk%):nhead=head
4785 for i=head to 27
4790 if massint%(i,vmspk%)< basemin% then
nhead=i:basemin%=massint%(i,vmspk%)
4795 next i
4800 basemin%=massint%(28,vmspk%):ntail=28
4805 for i=28 to tail
4810 if massint%(i,vmspk%)<= basemin% then
ntail=i:basemin%=massint%(i,vmspk%)
4815 next i
4820 head=nhead:tail=ntail
4830 if head < 12 then head=12
4840 if tail > 50 then tail=50
4850    return:rem--------------------------------
5000 REM auto-integrate ms
5010 PRINT "MS auto-integrate loop #";ll;" for
mass";massnum%(mspoint-1)
5020 DIM
AREA1(msPKPTR%),height%(mspkptr%),area4(mspkptr%),mselem%(m
spkptr%)
5022 dim vmloc%(mspk%),msbase%(mspkptr%)
5024 gosub 5200
5026 print
5030 if pkcount%=0 then print "no peak":goto 5120
5040 INPUT "Do you want to save integrals (y/n)"; YES$
5050 IF YES$="y" THEN GOSUB 5510:GOTO 5080
5060 IF YES$="n" THEN 5080
```

```
5070 GOTO 5040
5080 INPUT "Do you want to print integrals (y/n)"; YES$
5090 IF YES$="y" THEN GOSUB 5610:GOTO 5120
5100 IF YES$="n" THEN 5120
5110 GOTO 5080
5120 ERASE
AREA1,height%,area4,mselem%,vmloc%,msbase%:RETURN:REM-------
- - - - -
5200 rem ms auto-integrate
5210 mspk%=0:for i=0 to mspkptr%-1
5220 if massint%(0,i) <> (mspoint-1) then 5240
5230 vmloc%(mspk%)=massint%(1,i):mspk%=mspk%+1
5240 next i
5250 mk%=1:pkcount%=0
5260 FOR msPKi%=0 TO msPKPTR%-1:REM do it for every peak of this
mass
5270 if massint%(0,mspki%) <> (mspoint-1) then 5430
5280 mselem%(pkcount%)=massint%(1,mspki%)
5290 height%(pkcount%)=massint%(2,mspki%)
5300 vmspk%=mspki%
5310 gosub 4700
5320 mk%=mk%+1
5330 sum=0:sum4=0:area1(pkcount%)=0:area4(pkcount%)=0
5340 for i=head to tail
5350
AREA1(pkcount%)=AREA1(pkcount%)+(massint%(I,msPKi%)-baseline)
5360 area4(pkcount%)=area4(pkcount%)+(massint%(I+50,mspki%))
5370 NEXT I
5380 area4(pkcount%)=area4(pkcount%)/(tail-head+1)
5390 msBASE%(pkcount%)=baseline
5400 AREA1(pkcount%)=FIX(AREA1(pkcount%))
5410 PRINT pkcount%+1;" ";int
(AREA1(pkcount%)),int(area4(pkcount%))
5420 pkcount%=pkcount%+1
5430 NEXT msPKi%
5440    return:rem------------------------------
5510 REM saving integrals
5512
```

```
sname$=left$(flname$,4)+mid$(str$(massnum%(mspoint-1)+100),3)
5514  sname$=sname$+mid$(str$(ii),2)+".prn"
5520 OPEN "b:"+sname$ FOR OUTPUT AS # 2
5540 FOR msPKi%=0 TO pkcount%-1
5550  PRINT#2,  msPKi%+1,mselem%(mspki%)/2,height%(mspki%),;
5560  print#2,
int(AREA1(msPKi%)),msbase%(mspki%),int(area4(mspki%))
5570 NEXT msPKi%
5580  PRINT#2,
5590 CLOSE 2
5600    RETURN:REM--------------------------------
5610   REM   print   integral------------------------
5620 LPRINT FLNAME$;"    Loop #";ll ;"autointegrate
mass";massnum%(mspoint-1)
5630 LPRINT
5640 LPRINT "peak"; TAB(12);"at (sec.)";TAB(24);"height";
5642 LPRINT  TAB(36);"area";TAB(48);"baseline"TAB(58);"mass  4
avg.";
5643  lprint  tab(70);"pk.  ht."
5650 NL%=0:FOR msPK%=0 TO pkcount%-1
5660 LPRINT
msPK%+1;TAB(12);msELEM%(msPK%)/2;TAB(24);height%(mspk%);
5670 LPRINT
TAB(36);int(AREA1(msPK%));TAB(48);msBASE%(msPK%);
5671  lprint  TAB(58);int(area4(mspk%));
5672  lprint  tab(70);height%(mspk%)-msbase%(mspk%)
5674 NL%=NL%+1:IF NL%=5 THEN NL%=0:LPRINT
5680 NEXT msPK%
5682  lprint:lprint
5690    RETURN:REM--------------------------------
6000    rem    GC    plot-------------------------------
6002 print "Plot GC; Pen up ; type any key to continue"
6004  if inkey$="" then 6004
6010 xplot%=0:pbase%=gc%(0,0):yplot%=pbase%:gosub 6400
6012 print "Pen down ;type any key to continue"
6014  if inkey$="" then 6014
6020 for i=0 to gcpkptr%-1 :rem for every gcpeaks
6030 if xplot% >= (gcelem%(i)-30) then 6100
```

```
6040 yplot%=pbase%:gosub 6400
6050 xplot%=xplot%+1
6060 goto 6030
6100 selem%=xplot%-gcelem%(i)+30
6110 if selem% > 60 then 6200
6120 for i1=selem% to 60
6130 yplot%=gc%(i1,i):gosub 6400
6140 xplot%=xplot%+1
6150 next i1
6160 pbase%=gc%(0,i)
6170 for i1=0 to 60
6180 if pbase% > gc%(i1,i) then pbase%=gc%(i1,i)
6190 next i1
6200 next i
6390    return:rem--------------------------------
6400  rem  plot-------
6410 if xplot%<0 then xplot%=0
6420 if xplot%>4095 then xplot%=4095
6430 if yplot%<0 then yplot%=0
6440 if yplot%>4095 then yplot%=4095
6450 plot0%=255-yplot% and 255
6460 plot2%=255-xplot% and 255
6470  plot1%=255-(int(xplot%/256)*16+int(yplot%/256))
6480 out plotter+2,plot2%:out plotter+1,plot1%:out plotter,plot0%
6490 return:rem ----
6500   rem   MS   plot------------------------------
6502 print "Plot MS"massnum%(mspoint-1);" Pen up ; type any key to
continue"
6504 if inkey$="" then 6504
6510 xplot%=0:pbase%=0:yplot%=pbase%:gosub 6400
6512 print "Pen down ;type any key to continue"
6514 if inkey$="" then 6514
6520 for i=0 to mspkptr%-1 :rem for every mspeaks
6522 if massint%(0,i) <> (mspoint-1) then 6700
6524 if xplot%=0 then pbase%=massint%(3,i)
6530 if xplot% >= (massint%(1,i)-28+3) then 6600
6540 yplot%=pbase%:gosub 6400
6550 xplot%=xplot%+1
```

```
6560 goto 6530
6600  selem%=xplot%-massint%(1,i)+28
6610 if selem% > 52 then 6700
6620 for i1=selem% to 52
6630 yplot%=MASSINT%(i1,i):gosub 6400
6640 xplot%=xplot%+1
6650 next i1
6660 pbase%=massint%(3,i)
6670 for i1=3 to 52
6680 if pbase% > massint%(i1,i) then pbase%=massint%(i1,i)
6690 next i1
6700 next i
6990     return:rem--------------------------------
7000 rem auto align GC & MS data
7002 print "Check printer is ready, and there is enough paper"
7004 print "Type any key to continue"
7006 if inkey$="" then 7006
7008 Lprint "Auto align Loop";ii;" of file ";flname$:lprint
7010 dim
area(gcpkptr%),area1(mspkptr%),height%(mspkptr%),area4(mspkptr-
%)
7012 dim
mselem%(mspkptr%),vmloc%(mspkptr%),msbase%(mspkptr%),pkid%(g
cpkptr%)
7014 dim form(gcpkptr%-1,nummass%+1)
7022 print:print:print "Auto align routine"
7024 print "Now doing auto integrating GC"
7030 gosub 4200:rem auto gc integrate
7032 gosub 4400:rem print gc integrals
7040 print "Now aligning GC data with peak ID"
7050 idptr%=0
7060 for igc%=0 to gcpkptr%-1
7070 oskew%=30000
7072 if idptr%=94 then 7094
7080 skew%=abs(gcret%(idptr%)-gcelem%(igc%)/2)
7090 if skew% < oskew% then oskew%=skew%:idptr%=idptr%+1:goto
7072
7094 pkid%(igc%)=idptr%
```

```
7096  form(igc%,0)=idptr%:form(igc%,1)=area(igc%)
7098  rem  idptr%=idptr%-1
7110 next igc%
7120 for ims%=0 to nummass%-1
7130  mspoint=ims%+1
7140 print "auto integrating mass";massnum%(ims%)
7150 gosub 5200:rem ms integrate
7152 gosub 5610:rem print ms integrate
7160 if pkcount%=0 then print "no peak for this mass":goto 7700
7170 print "Aligning MS with GC data"
7180 igc%=0:for imspk%=0 to pkcount%-1
7190 oskew%=30000
7192 if igc% = gcpkptr% then 7240
7200
skew%=abs((mselem%(imspk%)-gcelem%(igc%))/2-msdel%(pkid%(igc
%)-1))
7210 if skew% < oskew% then oskew%=skew%:igc%=igc%+1:goto
7192
7220  form(igc%-1,ims%+2)=area1(imspk%)/area4(imspk%)
7230 igc%=igc%-1
7240 next imspk%
7700 next ims%
7701 lprint chr$(12);"Align Data for Loop";ii;" of file
";flname$:lprint
7702  lprint  "pkid";tab(15);"gcarea";
7703 for i=0 to numrnass%-1
7704  lprint  using  "########";massnum%(i);
7705  next  i:lprint
7706 for i=0 to gcpkptr%-1
7707  lprint  form(i,0);tab(5);gcelem%(i)/2;tab(15);form(i,1);tab(25)
7708 for iform=2 to nummass%+1:lprint using
"###.####";form(i,iform);:next
7709  lprint:next  i:lprint  chr$(12);
7710  sname$=left$(flname$,4)+"ag"+mid$(str$(ii),2)+".prn"
7720 open "b:"+sname$ for output as #2
7730  print#2,"pkid","gcarea",;
7740 for i=0 to nummass%-1
7750  print#2,using  "########";massnum%(i);
```

```
7760 next i
7762  print#2,
7770 for i=0 to gcpkptr%-1
7772  print#2,form(i,0),form(i,1),;
7780 for iform=2 to nummass%+1
7800  print#2,using  "###.####";form(i,iform);
7810 next iform
7820  print#2,
7830 next i
7840 close 2
7990 erase
area,area1,height%,area4,mselem%,vmloc%,msbase%,pkid%,form
7992    return:rem----------------------------------
9000  rem  error  handling  routine----------------
9010 if (err=76) and (erl=322) then resume next:rem not a dir
9020 if (err=53) and (erl=314) then resume next:rem no *.par
9030 if (err=53) and (erl=316) then resume next:rem no <DIR>
9040 if (err=53) and (erl=1210) then nummass%=0:goto 1200:rem no
"*.b"files
9090 open "tberr.txt" for output as 10
9100  print#10,erl,err
9110 close 10
9200 end:rem error end
9900 chdir "b:\":rem reset B drive working dir
9910 END:rem end program
```

Appendix II

Solution of the differential equations describing chain growth during Fischer-Tropsch Synthesis

The solutions for the differential equations that describe the chain growth model used in Chapter IV are derived in this Appendix. The starting point is the equations (eqn.1 and 3 in Chapter IV) for the fractional label incorporation in the monomer pool and in each of the C_n pools that are the precursors to the hydrocarbon products of chain length n. F_i is the fractional label content of pool i on the surface; τ_m is the lifetime of the monomer pool and τ is the lifetime of each of the C_n surface pools.

$$\frac{d\,F_m(t)}{dt} = \frac{1 - F_m(t)}{\tau_m} \qquad \ldots\ldots\ (1)$$

$$\frac{d\,F_n(t)}{dt} = \frac{\frac{(n-1)F_{n-1}(t) + F_m(t)}{n} - F_n(t)}{\tau} \qquad \text{for } n = 1,n \ldots\ldots\ (2)$$

Initial and Boundary conditions:

1. The F curves for the monomer and all the C_n pools start at 0 at t =0; i. e., $F_m(t=0) = F_i(t=0) = 0$, where i = 1, n.
2. At $t = \infty$, all F curves have reached steady state and are equal to 1.0

Laplace transforms of the first-order differential equations 1 and 2 are used to obtain solutions for $F_n(t)$. The Laplace transform

(L.T.) of $F_i(t)$ will be represented as $f_i(s)$.

Taking the L.T. of equation 1 gives:

$$sf_m(s) - F_m(0) = \frac{(\frac{1}{s} - f_m(s))}{\tau_m} \quad \ldots\ldots \quad (3)$$

From the initial condition for F_m, $F_m(0) = 0$; so this can be simplified to give

$$f_m(s) = \frac{1}{s\,\tau_m\,(s + \frac{1}{\tau_m})} \quad \ldots\ldots \quad (3')$$

This expression now yields the solution that

$$F_m(t) = 1.0 - \exp\,(\frac{-t}{\tau_m}) \quad \ldots\ldots \quad (4)$$

Taking the L.T. of equation 2 , we get:

$$sf_n(s) - F_n(0) = \frac{(n-1)}{n\tau} f_{n-1}(s) + \frac{1}{n\tau} f_b(s) - \frac{f_n(s)}{\tau} \quad \ldots\ldots \quad (5)$$

Again, using the initial condition that $F_n(0) = 0$, this yields:

$$f_n(s) = \frac{1}{n\tau(s+\frac{1}{\tau})}[f_b(s) + (n-1)\, f_{n-1}(s)] \quad \ldots\ldots \quad (5')$$

The L.T. of each pool depends on the L.T. of the previous pool in this set of recurring relationships as well as $f_b(s)$, except for the expression for $f_1(s)$ which only involves $f_b(s)$. From equation 5',

$$f_1(s) = \frac{1}{\tau(s+\frac{1}{\tau})}[f_b(s)] \quad \ldots\ldots \quad (6)$$

Now, this expression for $f_1(s)$ can be used with equation 5' to yield an expression for $f_2(s)$ and so on; ultimately, this results in the following expression for $f_n(s)$:

$$f_n(s) = \frac{f_b(s)}{n}\left[\frac{1}{\tau(s+\frac{1}{\tau})} + \frac{1}{\tau^2(s+\frac{1}{\tau})^2} + \ldots \frac{1}{\tau^n(s+\frac{1}{\tau})^n}\right] \quad \ldots\ldots (7)$$

The inverse transform of this expression will give the solution $F_n(t)$. Since all the terms can be expressed by a general expression for the ith term, the ith term of the above expression is inverted; the summation of this for all terms from i=1 to i=n then yields $F_n(t)$. The ith term in the expansion is:

$$f_i(s) = \frac{f_b(s)}{n}\left[\frac{1}{\tau^i(s+\frac{1}{\tau})^i}\right]$$

To get the L^{-1} or inverse L.T. of this term, use is made of the convolution theorem (1), which states:

If $f(s) = g(s)h(s)$, and $L^{-1}[g(s)] = G(t)$ and $L^{-1}[h(s)] = H(t)$,

$$F(t) = \int_0^t G(u)H(t-u)du \quad \ldots\ldots (8)$$

(where $F(t) = L^{-1}[f(s)]$)

Also, from L.T. tables, the following inverse transform is obtained:

$$L^{-1}\left[\frac{1}{(s-a)^i}\right] = \frac{1}{(i-1)!}t^{i-1}\exp(at) \quad \ldots\ldots (9)$$

Using eqns. 4, 8 and 9,

$$L^{-1}[\text{ith term}] = \frac{1}{n\tau^i(i-1)!}\int_0^t u^{i-1}\exp(\frac{-u}{\tau})(1-\exp(\frac{-(t-u)}{\tau_m}))du \quad \ldots\ldots (10)$$

The expression can be evaluated using integral tables (2) to give:

$$L^{-1}[\text{ith term}] = \frac{1}{n}\{1 + \exp(\frac{-t}{\tau_m})(-1)^{i-1}(\frac{\tau_m}{\tau-\tau_m})^i + \frac{\exp(\frac{-t}{\tau})}{\tau^i}\sum_{r=0}^{i-1}\frac{t^{i-r-1}\tau^{r+1}}{(i-r-1)!}((\frac{-\tau_m}{\tau-\tau_m})^{r+1}-1)\} \quad \ldots\ldots (11)$$

From eqn. 11, $F_n(t)$ can be evaluated:

$$F_n(t) = \sum_{i=1}^{n} [L^{-1}(\text{ith term})]$$

$$F_n(t) = 1.0 + \left(\frac{\exp(-\frac{t}{\tau_m})}{n}\right)\sum_{i=1}^{n} (-1)^{i-1}\left(\frac{\tau_m}{\tau-\tau_m}\right)^i$$

$$+ \left(\frac{\exp(-\frac{t}{\tau})}{n}\right)\sum_{i=1}^{n} \frac{1}{\tau^i}\sum_{r=0}^{i-1} \frac{t^{i-r-1}}{(i-r-1)!}\tau^{r+1}\left(\left(-\frac{\tau_m}{\tau-\tau_m}\right)^{r+1} - 1\right) \quad \ldots\ldots \quad (12)$$

Equation 12 can now be used to generate expressions for any desired value of the carbon number, n.

REFERENCES

1. Jenson, V. G., and Jeffreys, G. V., "Mathematical Methods in Chemical Engineering", Second Edition, Academic Press, 1977.
2. "CRC Handbook of Chemistry and Physics", R. C. Weast (ed.), 63rd edition, CRC Press Inc., Boca Raton, Florida, 1982.

Appendix III

Chain-growth model parameter estimation

This Appendix describes the chain growth model fitting program used for parameter estimation from the model presented in Chapter 4. The program is written in FORTRAN and has been run on the Main Frame IBM 3081.

3.1 Program Minimum

This is a program that can provide parameter estimates for a function of several variables, by minimizing the function using a quasi-Newton method. The user must provide a subroutine to calculate the function and its gradient (first partial derivatives with respect to the variables of interest.) This function is the objective function, which consists of the summation of the square of the difference of the experimental data points and the function evaluated at these points using the current estimates of the solution vector. The user initially inputs the number of variables (2 in our case) and an estimate or initial guess of the solution vector for these variables, and the original experimental data points. The program iteratively searches for the parameter values that minimize the provided objective function. Finally, the program output gives the final solution vector, and the function values that correspond to the experimental data points, as well as a termination indicator that indicates whether convergence occurred.

'MINIMUM' is the name of the FORTRAN main program used here; it is a general error minimization routine that can be used for any user-supplied function. The Subroutine 'SUMDATA' provides the chain growth model objective function and its partial derivatives with respect to the two parameters, τ and τ_m. The Subroutine 'FUNC' that is called by subroutine 'SUMDATA' actually evaluates the $F_n(t, n, \tau, \tau_m)$ that is obtained as a solution for the chain growth model presented in Chapter 4 at input values of these four parameters. If a different solution function needs to be used, it can be modified in this subroutine. The user also inputs as data the F (rise) curves obtained from the transient response isotopic tracer experiments described in Chapter 4. The input requires the carbon number of the F curve, initial guesses of τ and τ_m and finally, t_i, $F^{expt}(t_i)$ pairs. A listing of the program follows.

PROGRAM MINIMUM

```
C=======================================================================MIN00010
                                                                       MIN00020
      Program Minimum                                                  MIN00030
                                                                       MIN00040
      implicit none                                                    MIN00050
                                                                       MIN00060
      INTEGER   INPUTUNIT,MAXDATA,N                                    MIN00070
      integer   nvar,ihess,maxf,irhess,iwhess,iprint                   MIN00080
      INTEGER   IERR,ISWTCH,ITER,NFCALL,I                              MIN00090
      parameter (nvar=2,ihess=.5*nvar*(nvar+1))                        MIN00100
      real*8    tolx,tolg,sprec,extbnd,rfn,objf,histry,                MIN00110
     2          x(nvar),g(nvar),solvec(nvar),srhvec(nvar),grdvec(nvar)MIN00120
      REAL*8    HESS(IHESS),SCRVEC(NVAR),NORM,T(100),FEXP(100)         MIN00130
      real*8    fn,deltau,deltaub                                      MIN00140
      LOGICAL   ALWAYSTRUE                                             MIN00150
c external function calls for psi and phi                              MIN00160
                                                                       MIN00170
      external sumdata                                                 MIN00180
                                                                       MIN00190
c initialize input parameters for the minimization routine             MIN00200
                                                                       MIN00210
      data maxf,tolx,tolg,sprec,extbnd,irhess,iwhess,iprint,ierr,      MIN00220
     2     iswtch,rfn/500,1d-6,1d-10,.1,2.,0,0,0,0,2,1d-6/             MIN00230
                                                                       MIN00240
      data AlwaysTrue/.true./                                          MIN00250
      data inputunit/20/                                               MIN00260
      DATA OBJF/0.0D0/                                                 MIN00270
                                                                       MIN00280
      COMMON /EXPTDATAR/T,FEXP                                         MIN00290
      common /exptdataI/MaxData                                        MIN00300
      COMMON /CARBON/N                                                 MIN00310
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc MIN00320
                                                                       MIN00330
C     OPEN (UNIT=INPUTUNIT,FILE='INPUT.DAT',STATUS='UNKNOWN')          MIN00340
C     OPEN(UNIT=INPUTUNIT)                                             MIN00350
c initial guess on parameters                                          MIN00360
                                                                       MIN00370
C     READ(INPUTUNIT,*) X                                              MIN00380
C     WRITE(6,*)X                                                      MIN00390
      DO 1 I=1,NVAR                                                    MIN00400
      READ(20,*)X(I)                                                   MIN00410
 1    CONTINUE                                                         MIN00420
c data points used in fitting function                                 MIN00430
                                                                       MIN00440
C     I=0                                                              MIN00450
C     DO WHILE (ALWAYSTRUE)                                            MIN00460
                                                                       MIN00470
C        I=I+1                                                         MIN00480
C        READ(INPUTUNIT,*,END=999,ERR=999)T(I),FEXP(I)                 MIN00490
C                                                                      MIN00500
C     ENDDO                                                            MIN00510
C                                                                      MIN00520
C99   CLOSE (UNIT=INPUTUNIT)                                           MIN00530
```

```
C       MAXDATA = I-1                                                    MIN00540
C       MAXDATA IS THE # OF DATA POINTS;N IS THE CARBON # OF DATA        MIN00550
        READ(20,*)N,MAXDATA                                              MIN00560
        DO 2 I=1,MAXDATA                                                 MIN00570
        READ(20,*)T(I),FEXP(I)                                           MIN00580
C       WRITE(6,*)'T=',T(I),'FEXP',FEXP(I)                               MIN00590
 2      CONTINUE                                                         MIN00600
        do i=1,ihess                                                     MIN00610
           hess(i)=0.0d0                                                 MIN00620
        enddo                                                            MIN00630
                                                                         MIN00640
        call gqbfgs(nvar,x,iswtch,maxf,iter,iprint,tolx,tolg,rfn,        MIN00650
     2       sprec,extbnd,objf,ihess,hess,g,ierr,nfcall,srhvec,          MIN00660
     3       solvec,grdvec,scrvec,histry,irhess,iwhess,sumdata)          MIN00670
                                                                         MIN00680
        WRITE(22,*)'OBJF: ',OBJF                                         MIN00690
                                                                         MIN00700
        NORM = 0.0D0                                                     MIN00710
        do i = 1,nvar                                                    MIN00720
           norm = norm + g(i)**2                                         MIN00730
        enddo                                                            MIN00740
        NORM =DSQRT(NORM)                                                MIN00750
                                                                         MIN00760
        WRITE(22,*)'FINAL X: ',SOLVEC                                    MIN00770
        WRITE(22,*)'GRADIENT: '                                          MIN00780
        WRITE(22,*)(G(I),I=1,NVAR)                                       MIN00790
        WRITE(22,*)'NORM: ',NORM                                         MIN00800
        WRITE(22,*)'CARBON#',N                                           MIN00810
        WRITE(22,*)'ITER: ',ITER                                         MIN00820
        WRITE(22,*)'IERR: ',IERR                                         MIN00830
        WRITE(22,*)'NFCALL: ',NFCALL                                     MIN00840
        WRITE(22,*)                                                      MIN00850
                                                                         MIN00860
        WRITE(23,*) '"MODEL',N,'".'                                      MIN00870
        do i = 1,MaxData                                                 MIN00880
           call FUNC(T(i),FN,DELTAU,DELTAUB,X,MaxData)                   MIN00890
           WRITE(23,*) T(I),FN                                           MIN00900
        enddo                                                            MIN00910
        WRITE(23,*)                                                      MIN00920
        WRITE(23,*) '"EXP',N,'".'                                        MIN00930
        DO I = 1,MAXDATA                                                 MIN00940
           WRITE(23,*) T(I),FEXP(I)                                      MIN00950
        ENDDO                                                            MIN00960
                                                                         MIN00970
        stop                                                             MIN00980
        end                                                              MIN00990
                                                                         MIN01000
                                                                         MIN01010
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc MIN01020
c note: all real variables in this routine are double precision          MIN01030
c so make sure that any variable transfered into the routine is          MIN01040
c double precision (64 bit)                                              MIN01050
                                                                         MIN01060
      subroutine gqbfgs(nvar,x,iswtch,maxf,iter,iprint,tolx,tolg,        MIN01070
     1           rfn,sprec,extbnd,objf,ihess,hess,g,ierr,nfcall,srhvec, MIN01080
     2           solvec,grdvec,scrvec,histry,irhess,iwhess,funct)       MIN01090
```

```
c                                                                       MIN01100
      implicit real*8 (a-h,o-z)                                         MIN01110
c                                                                       MIN01120
      dimension  x(nvar),hess(ihess),g(nvar),srhvec(nvar),              MIN01130
     1           solvec(nvar),grdvec(nvar),scrvec(nvar)                 MIN01140
c                                                                       MIN01150
      external funct                                                    MIN01160
c                                                                       MIN01170
c     this subroutine minimizes a function of several variables         MIN01180
c     objf(x(1),x(2),...,x(nvar))  using a quasi-newton method          MIN01190
c     optionally employing the  dfp  and  bfgs  updating formulas.      MIN01200
c     the user must provide a subroutine to calculate objf(x) and       MIN01210
c     its gradient (first partial derivative) vector g(x).              MIN01220
c                                                                       MIN01230
c     ***************************************************************   MIN01240
c     this routine invokes the package modules  search  and  uphess     MIN01250
c     and the user supplied subroutine  funct.                          MIN01260
c     ***************************************************************   MIN01270
c                                                                       MIN01280
c     on input:                                                         MIN01290
c                                                                       MIN01300
c        nvar  is the number of variables. it is also the dimension of  MIN01310
c           the vectors  x, g, srhvec, solvec, grdvec  and  scrvec.     MIN01320
c                                                                       MIN01330
c        x  contains an estimate of the solution vector                 MIN01340
c           (x(1),x(2),...,x(nvar)).                                    MIN01350
c                                                                       MIN01360
c        iswtch  is a parameter set equal to  k  which selects the      MIN01370
c           formula used to update the approximation to the hessian     MIN01380
c           inverse. for                                                MIN01390
c                                                                       MIN01400
c           k = 1 - the  dfp  update,                                   MIN01410
c           k = 2 - the  bfgs  update.                                  MIN01420
c                                                                       MIN01430
c           the  bfgs  update is recommended.                           MIN01440
c                                                                       MIN01450
c        maxf  is the limit on the number of calls to the function      MIN01460
c           evaluation routine  funct.                                  MIN01470
c                                                                       MIN01480
c        funct  is a user supplied subroutine to evaluate  objf(x)      MIN01490
c           and the components of the gradient  g(x)  at the estimate   MIN01500
c           x(i), i = 1,2,...,nvar. Decternal in calling        LAR01390MIN01510
c           routine.                                                    MIN01520
c                                                                       MIN01530
c        tolx,tolg are the accuracies required in the solution, i.e. a  MIN01540
c           normal return from the routine occurs if the difference     MIN01550
c           between the components of two successive estimates of the   MIN01560
c           solution are not greater than  max(tolx*abs(x(i)),tolx)     MIN01570
c           for all  i, and the l2 norm of the gradient is not greater  MIN01580
c           than  tolg.                                                 MIN01590
c                                                                       MIN01600
c        rfn  is an estimate of the expected reduction in  objf(x).     MIN01610
c           this estimate is used only on the first iteration so an     MIN01620
c           order of magnitude estimate will suffice. the information   MIN01630
c           can be provided in the following ways depending upon the    MIN01640
c           value of  rfn. for                                          MIN01650
```

```
c                                                                       MIN01660
c          rfn .gt. 0.0 - the setting of  rfn  itself will be taken     MIN01670
c                         as the expected reduction in  objf(x),        MIN01680
c                                                                       MIN01690
c          rfn = 0.0    - it is assumed that an estimate of the         MIN01700
c                         minimum value of  objf(x)  has been set       MIN01710
c                         in the argument  objf, and the expected       MIN01720
c                         reduction in  objf(x)  will be computed       MIN01730
c                         as (initial function value) minus objf,       MIN01740
c                                                                       MIN01750
c          rfn .lt. 0.0 - a multiple  abs(rfn)  of the modulus of       MIN01760
c                         the initial function value will be taken      MIN01770
c                         as the expected reduction.                    MIN01780
c                                                                       MIN01790
c       sprec  is the accuracy required in the linear search technique  MIN01800
c          invoked by  gqbfgs, i.e. a point  xm  is accepted as the     MIN01810
c          minimum along the search direction if the ratio of the       MIN01820
c          directional derivative at  xm  over the directional          MIN01830
c          derivative at the initial point is not greater than  sprec.  MIN01840
c                                                                       MIN01850
c          the setting  0.100  is recommended.                          MIN01860
c                                                                       MIN01870
c       extbnd  is the upper bound on the multiplicative increase in    MIN01880
c          the search scang the extrapolation phase of the              MIN01890
c          linear search technique.                                     MIN01900
c                                                                       MIN01910
c          the setting  2.000  is recommended.                          MIN01920
c                                                                       MIN01930
c       objf  contains an estimate of the minimum value of  objf(x)     MIN01940
c          if  rfn = 0.0. otherwise it is only an output parameter.     MIN01950
c                                                                       MIN01960
c       ihess  is a parameter set equal to the dimension of  hess       MIN01970
c          which is at least  nvar*(nvar+1)/2.                          MIN01980
c                                                                       MIN01990
c       histry  is a dummy parameter.                                   MIN02000
c                                                                       MIN02010
c     on output                                                         MIN02020
c                                                                       MIN02030
c                                                                       MIN02040
c       x  contains the best available estimate of the solution vector. MIN02050
c                                                                       MIN02060
c       objf  contains the function value at  x.                        MIN02070
c                                                                       MIN02080
c                                                                       MIN02090
c       hess  is an array of dimension  ihess  which contains the       MIN02100
c          upper triangle of the most recent approximation to the       MIN02110
c          hessian inverse stored row-wise.                             MIN02120
c                                                                       MIN02130
c       g  contains the components of the gradient at  x.               MIN02140
c                                                                       MIN02150
c       ierr  is a parameter set equal to  k  which gives the following MIN02160
c          termination indications                                      MIN02170
c                                                                       MIN02180
c          normal termination,                                          MIN02190
c            k = 0,                                                     MIN02200
c          intermediate termination,                                    MIN02210
```

```
c             k = -n(n any integer) - user termination,                  MIN02220
c             k = 1 - failure to converge in  maxf  calls of  funct,     MIN02230
c             k = 2 - linear search technique indicates that it is       MIN02240
c                     likely that no minimum exists,                     MIN02250
c                                                                        MIN02260
c        nfcall  is the number of calls to  funct.                       MIN02270
c                                                                        MIN02280
c        srhvec  contains the current search direction vector.           MIN02290
c                                                                        MIN02300
c        solvec  contains the current solution vector.                   MIN02310
c                                                                        MIN02320
c        grdvec  contains the current gradient vector.                   MIN02330
c                                                                        MIN02340
c        scrvec  is a scratch vector.                                    MIN02350
c                                                                        MIN02360
c     written by k. e. hillstrom, march, 1976.                           MIN02370
c                                                                        MIN02380
c                                                                        MIN02390
c     initialize the following parameters                                MIN02400
c                                                                        MIN02410
c       ierr   - the termination indicator                               MIN02420
c       nfcall - the number of calls to  funct                           MIN02430
c       redfcn - the initial predicted reduction in  objf                MIN02440
c       iter   - the current iteration number                            MIN02450
c                                                                        MIN02460
      ierr = 0                                                           MIN02470
      iter=0                                                             MIN02480
      nfcall = 1                                                         MIN02490
      temp = objf                                                        MIN02500
c                                                                        MIN02510
c     ********************************************************************MIN02520
      call funct(nvar,x,objf,g)                                          MIN02530
c     ********************************************************************MIN02540
c                                                                        MIN02550
      sqgrad=0.0                                                         MIN02560
      do 220 iii=1,nvar                                                  MIN02570
  220     sqgrad=sqgrad+g(iii)**2                                        MIN02580
c                                                                        MIN02590
      if (ierr .lt. 0) go to 410                                         MIN02600
      redfcn = rfn                                                       MIN02610
      if (rfn .eq. 0.0) redfcn = objf - temp                             MIN02620
      if (rfn .lt. 0.0) redfcn = abs(redfcn * objf)                      MIN02630
      if (redfcn .le. 0.0) redfcn = 1.0                                  MIN02640
c                                                                        MIN02650
c    read initial estimate of hessian inverse, if desired               MIN02660
      if(irhess.eq.0) go to 200                                          MIN02670
      read(11,202) (hess(ii),ii=1,ihess)                                 MIN02680
  202   format(5e16.9)                                                   MIN02690
      go to 300                                                          MIN02700
c                                                                        MIN02710
c     begin the quasi-newton process by initializing the approximation   MIN02720
c       to the hessian inverse to unity                                  MIN02730
c                                                                        MIN02740
  200 if(iprint.ne.0) write(10,201)                                      MIN02750
  201 format(//2x,'>>>> quasi-newton procedure started, with search',    MIN02760
     1       ' direction set to -g')                                     MIN02770
```

```
      k = 1 + nvar * (nvar+1) / 2                                    MIN02780
c                                                                    MIN02790
      do 210 i = 1, nvar                                             MIN02800
c                                                                    MIN02810
         do 205 j = 1, i                                             MIN02820
            k = k - 1                                                MIN02830
            hess(k) = 0.0                                            MIN02840
  205    continue                                                    MIN02850
c                                                                    MIN02860
         hess(k) = 1.0                                               MIN02870
  210 continue                                                       MIN02880
c                                                                    MIN02890
  300 if(iprint.eq.0) go to 301                                      MIN02900
      euclid=sqrt(sqgrad)                                            MIN02910
      if(iprint.ge.20.and.mod(iter,10).ne.0) go to 308               MIN02920
      write(6,307) iter,nfcall,objf,euclid                           MIN02930
  307 format(1x,'iteration:',i5,26x,'function evaluation:',i6,       MIN02940
     1       /1x,'objective function:',e20.13,2x,'gradient norme:',  MIN02950
     2        e20.13)                                                MIN02960
  308 if(mod(iter,iprint).ne.0) go to 301                            MIN02970
      write(10,302) iter,nfcall                                      MIN02980
  302 format(////2x,'iteration no ',i5//2x,'number of function and ', MIN02990
     1       'gradient evaluations = ',i5//2x,'parameter values')    MIN03000
      write(10,303) (j,x(j),j=1,nvar)                                MIN03010
  303 format(/3(2x,'x(',i4,') = ',e16.8))                            MIN03020
      write(10,304) objf                                             MIN03030
  304 format(//2x,'function value objf = ',e16.8///2x,'gradient')    MIN03040
      write(10,306) (j,g(j),j=1,nvar)                                MIN03050
  306 format(/3(2x,'g(',i4,') = ',e16.8))                            MIN03060
  301 iter=iter+1                                                    MIN03070
c                                                                    MIN03080
c     begin an iteration by saving the current best estimate of the  MIN03090
c       function and the solution and gradient vectors.              MIN03100
c                                                                    MIN03110
      do 310 i = 1, nvar                                             MIN03120
         solvec(i) = x(i)                                            MIN03130
         grdvec(i) = g(i)                                            MIN03140
  310 continue                                                       MIN03150
c                                                                    MIN03160
      tobjf = objf                                                   MIN03170
c                                                                    MIN03180
c     calculate the search direction vector in  srhvec  and the      MIN03190
c       directional derivative in  dirdev                            MIN03200
c                                                                    MIN03210
      do 340 i = 1, nvar                                             MIN03220
         ij = i                                                      MIN03230
         z = 0.0                                                     MIN03240
c                                                                    MIN03250
         do 330 j = 1, nvar                                          MIN03260
            z = z - g(j) * hess(ij)                                  MIN03270
            if (j .ge. i) go to 325                                  MIN03280
            ij = ij + nvar - j                                       MIN03290
            go to 330                                                MIN03300
  325       ij = ij + 1                                              MIN03310
  330    continue                                                    MIN03320
c                                                                    MIN03330
```

```
         srhvec(i) = z                                                  MIN03340
  340 continue                                                          MIN03350
c                                                                       MIN03360
      dirdev = 0.0                                                      MIN03370
c                                                                       MIN03380
      do 350 i = 1, nvar                                                MIN03390
         dirdev = dirdev + srhvec(i) * g(i)                             MIN03400
  350 continue                                                          MIN03410
c                                                                       MIN03420
c     if the directional derivative  dirdev  is .gt. 0, there is no     MIN03430
c       guarantee that a search in the  w  direction  will result in a  MIN03440
c       smaller  objf. therefore, the  quasi-newton  process is         MIN03450
c       restarted at the current estimate of the solution with  srhvec  MIN03460
c       set to  -g.                                                     MIN03470
c                                                                       MIN03480
      if (dirdev .gt. 0.0) go to 200                                    MIN03490
      if (dirdev .eq. 0.0) go to 500                                    MIN03500
c                                                                       MIN03510
c     compute the initial search scaha  and conduct the                 MIN03520
c       linear search by means of a call to  search                     MIN03530
c                                                                       MIN03540
      alpha = -2.0 * redfcn / dirdev                                    MIN03550
      if (alpha .gt. 1.0) alpha = 1.0                                   MIN03560
      redfcn = objf                                                     MIN03570
c                                                                       MIN03580
c     ****************************************************************  MIN03590
      call search(nvar,x,g,srhvec,objf,alpha,dirdev,sprec,              MIN03600
     1            extbnd,nfcall,scrvec,ierr,funct,maxf)                 MIN03610
c     ****************************************************************  MIN03620
c                                                                       MIN03630
c     test for abnormal termination                                     MIN03640
c                                                                       MIN03650
      if (ierr .lt. 0) go to 500                                        MIN03660
      if (nfcall .ge. maxf) go to 400                                   MIN03670
      if ((alpha .lt. 1.0e-20) .or.                                     MIN03680
     1    (alpha .gt. 1.0e20)) go to 410                                MIN03690
c                                                                       MIN03700
c     test for convergence                                              MIN03710
c                                                                       MIN03720
      sqgrad = 0.0                                                      MIN03730
      iconv = 0                                                         MIN03740
c                                                                       MIN03750
      do 360 i = 1, nvar                                                MIN03760
         temp = alpha * srhvec(i)                                       MIN03770
         sqgrad = sqgrad + g(i) * g(i)                                  MIN03780
         t = tolx * abs(x(i))                                           MIN03790
         if (t .le. tolx) t = tolx                                      MIN03800
         if (abs(temp) .gt. t) iconv = 1                                MIN03810
  360 continue                                                          MIN03820
c                                                                       MIN03830
      if (sqgrad .gt. tolg*tolg) iconv = 1                              MIN03840
      if (sqgrad .eq. 0.0) iconv = 0                                    MIN03850
      if (iconv .eq. 0) go to 500                                       MIN03860
c                                                                       MIN03870
c     the linear search technique has located a minimum. call  uphess   MIN03880
c       to update the approximation to the hessian inverse using the    MIN03890
```

```
c       dfp  or  bfgs  updating formulas                                MIN03900
c                                                                       MIN03910
c     ***************************************************************   MIN03920
      call uphess(nvar,x,g,ihess,hess,solvec,grdvec,scrvec,iswtch,iexit)MIN03930
c     ***************************************************************   MIN03940
c                                                                       MIN03950
c     if the update is not successful the quasi-newton process is       MIN03960
c       restarted with a descent step at the current estimate of the    MIN03970
c       solution                                                        MIN03980
c                                                                       MIN03990
      redfcn = redfcn - objf                                            MIN04000
      if (iexit .ne. 0) go to 200                                       MIN04010
c                                                                       MIN04020
c     now start a new iteration                                         MIN04030
c                                                                       MIN04040
c  write the current estimate of the hessian inverse, if desired.       MIN04050
      if(iwhess.eq.0) go to 300                                         MIN04060
      write(12,202) (hess(ii),ii=1,ihess)                               MIN04070
c                                                                       MIN04080
      go to 300                                                         MIN04090
c                                                                       MIN04100
c     error return because there have been at least  maxf  calls of     MIN04110
c       funct                                                           MIN04120
c                                                                       MIN04130
  400 ierr = 1                                                          MIN04140
      go to 450                                                         MIN04150
c                                                                       MIN04160
c     error return because linear search technique indicates that it is MIN04170
c       likely that no minimum exists                                   MIN04180
c                                                                       MIN04190
  410 ierr = 2                                                          MIN04200
c                                                                       MIN04210
  450 do 455 i = 1, nvar                                                MIN04220
         x(i) = solvec(i)                                               MIN04230
         g(i) = grdvec(i)                                               MIN04240
  455 continue                                                          MIN04250
c                                                                       MIN04260
      objf = tobjf                                                      MIN04270
  500 return                                                            MIN04280
c                                                                       MIN04290
      end                                                               MIN04300
                                                                        MIN04310
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc MIN04320
      subroutine uphess(n,x,g,ih,h,solvec,grdvec,scrvec,iswtch,iexit)   MIN04330
c                                                                       MIN04340
      implicit real*8 (a-h,o-z)                                         MIN04350
c                                                                       MIN04360
      dimension  x(n),g(n),h(ih),solvec(n),                             MIN04370
     1           grdvec(n),scrvec(n)                                    MIN04380
c                                                                       MIN04390
c     this subroutine updates an approximation to the hessian inverse   MIN04400
c       using the  dfp  or  bfgs  formula                               MIN04410
c                                                                       MIN04420
c     on input                                                          MIN04430
c                                                                       MIN04440
c        n  is the dimension of the vectors  x, g, solvec, grdvec  and  MIN04450
```

```
c           scrvec.                                                    MIN04460
c                                                                      MIN04470
c        x  contains an estimate of the solution vector.               MIN04480
c                                                                      MIN04490
c        g  contains the components of the gradient corresponding to   MIN04500
c           the  x  vector.                                            MIN04510
c                                                                      MIN04520
c        ih  is a parameter set equal to the dimension of  h  which is MIN04530
c           at least  n*(n+1)/2.                                       MIN04540
c                                                                      MIN04550
c        h  is an array of dimension  ih  which contains the upper     MIN04560
c           triangle of an approximation to the hessian inverse stored MIN04570
c           by rows.                                                   MIN04580
c                                                                      MIN04590
c        solvec  contains the current solution vector.                 MIN04600
c                                                                      MIN04610
c        grdvec  contains the current gradient vector.                 MIN04620
c                                                                      MIN04630
c        iswtch  is a parameter set equal to  k  which selects the     MIN04640
c           updating formula. for                                      MIN04650
c                                                                      MIN04660
c           k = 1 - the  dfp  formula is used,                         MIN04670
c           k = 2 - the  bfgs  formula is used.                        MIN04680
c                                                                      MIN04690
c     on output                                                        MIN04700
c                                                                      MIN04710
c        iexit  is a parameter set equal to  k  which indicates the    MIN04720
c           following. for                                             MIN04730
c                                                                      MIN04740
c           k = 0 - the update was successful,                         MIN04750
c           k = 1 - the update failed due to zero divisors.            MIN04760
c                                                                      MIN04770
c        h  contains the updated approximation to the hessian inverse  MIN04780
c           if  iexit = 0.                                             MIN04790
c                                                                      MIN04800
c        scrvec  is a scratch vector.                                  MIN04810
c                                                                      MIN04820
c     written by k. e. hillstrom, march, 1976.                         MIN04830
c                                                                      MIN04840
c                                                                      MIN04850
c     initialize the exit indicator  iexit                             MIN04860
c                                                                      MIN04870
      iexit = 0                                                        MIN04880
c                                                                      MIN04890
c     calculate the solution and gradient difference vectors from two  MIN04900
c       consecutive iterations. from this section on                   MIN04910
c                                                                      MIN04920
c       solvec - contains  delta, the solution difference vector       MIN04930
c       grdvec - contains  gamma, the gradient difference vector       MIN04940
c                                                                      MIN04950
  100 do 110 i = 1, n                                                  MIN04960
        solvec(i) = x(i) - solvec(i)                                   MIN04970
        grdvec(i) = g(i) - grdvec(i)                                   MIN04980
  110 continue                                                         MIN04990
c                                                                      MIN05000
c     calculate  z = (gamma transpose) * delta and alpha =             MIN05010
```

```
c       (gamma transpose) * (hessian inverse) * gamma  occuring as      MIN05020
c       denominators in the  dfp  formula. from this section on         MIN05030
c                                                                       MIN05040
c       h      - contains the approximation to the hessian inverse      MIN05050
c       scrvec - contains the successive elements of  (gamma transpose) MIN05060
c                * (hessian inverse)                                    MIN05070
c                                                                       MIN05080
      z = 0.0                                                           MIN05090
      alpha = 0.0                                                       MIN05100
c                                                                       MIN05110
      do 130 i = 1, n                                                   MIN05120
         wt = grdvec(i)                                                 MIN05130
         z = z + wt * solvec(i)                                         MIN05140
         k = i                                                          MIN05150
         wt = 0.0                                                       MIN05160
c                                                                       MIN05170
         do 120 j = 1, n                                                MIN05180
            wt = wt + grdvec(j) * h(k)                                  MIN05190
            if (j .ge. i) go to 115                                     MIN05200
            k = k + n - j                                               MIN05210
            go to 120                                                   MIN05220
  115       k = k + 1                                                   MIN05230
  120    continue                                                       MIN05240
c                                                                       MIN05250
         alpha = alpha + wt * grdvec(i)                                 MIN05260
         scrvec(i) = wt                                                 MIN05270
  130 continue                                                          MIN05280
c                                                                       MIN05290
c     error exit if the  dfp  or  bfgs  formula breaks down due to zero MIN05300
c       divisors  z  and/or  alpha                                      MIN05310
c                                                                       MIN05320
      if ((z .eq. 0.0) .or.                                             MIN05330
     1    (alpha .eq. 0.0 .and. iswtch .eq. 1)) go to 200               MIN05340
c                                                                       MIN05350
c     update the approximation to the hessian inverse using the  dfp    MIN05360
c       or  bfgs  updating formula                                      MIN05370
c                                                                       MIN05380
      k = 1                                                             MIN05390
c                                                                       MIN05400
      do 160 i = 1, n                                                   MIN05410
c                                                                       MIN05420
         do 150 j = i, n                                                MIN05430
            if (iswtch .eq. 1) go to 135                                MIN05440
            h(k) = h(k) - (solvec(i) * scrvec(j) + scrvec(i) *          MIN05450
     1             solvec(j)) / z + (1.0 + alpha / z) * (solvec(i) *   MIN05460
     2             solvec(j) / z)                                       MIN05470
            go to 140                                                   MIN05480
  135       h(k) = h(k) + solvec(i) * solvec(j) / z - scrvec(i) *      MIN05490
     1             scrvec(j) / alpha                                    MIN05500
  140       k = k + 1                                                   MIN05510
  150    continue                                                       MIN05520
c                                                                       MIN05530
  160 continue                                                          MIN05540
c                                                                       MIN05550
      go to 300                                                         MIN05560
c                                                                       MIN05570
```

```
c     error return due to zero divisors in the updating formula          MIN05580
c                                                                        MIN05590
  200 iexit = 1                                                          MIN05600
  300 return                                                             MIN05610
c                                                                        MIN05620
      end                                                                MIN05630
                                                                         MIN05640
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc MIN05650
      subroutine search(n,x,g,s,f,alpha,dirdev,sprec,extbnd,             MIN05660
     1                  nfcall,w,ierr,funct,maxf)                        MIN05670
c                                                                        MIN05680
      implicit real*8 (a-h,o-z)                                          MIN05690
c                                                                        MIN05700
      dimension x(n),g(n),s(n),w(n)                                      MIN05710
c                                                                        MIN05720
c       this subroutine obtains an estimate of the solution              MIN05730
c     xm = x0 + alpha * s  which minimizes  f  by means of a linear      MIN05740
c     search in the  s  direction.                                       MIN05750
c                                                                        MIN05760
c     on input                                                           MIN05770
c                                                                        MIN05780
c        n  is the number of variables. it is also the dimension of the  MIN05790
c           vectors  x, g  and  s.                                       MIN05800
c                                                                        MIN05810
c        x  contains an estimate of the solution vector                  MIN05820
c           (x0(1),x0(2),...,x0(n)).                                     MIN05830
c                                                                        MIN05840
c        s  contains the search direction vector.                        MIN05850
c                                                                        MIN05860
c        f  contains the objective function  f(x).                       MIN05870
c                                                                        MIN05880
c        funct  is a user supplied routine to evaluate  f(x)  and the    MIN05890
c           components of the gradient  g(x)  at the estimate in  x.     MIN05900
c                                                                        MIN05910
c        alpha  is  the initial step sca                    LAR05800MIN05920
c                                                                        MIN05930
c        dirdev  is the directional derivative at x.                     MIN05940
c                                                                        MIN05950
c        sprec  is the accuracy required in the search, i.e. a point  xmMIN05960
c           is accepted as the minimum along direction  s  if the ratio  MIN05970
c           of the directional derivative at  xm  over the directional   MIN05980
c           derivative at  x0  is not greater than  sprec.               MIN05990
c                                                                        MIN06000
c        extbnd  is the upper bound on the multiplicative increase in    MIN06010
c           alpha  during extrapolaticn.                                 MIN06020
c                                                                        MIN06030
c        nfcall  is the number of calls to the function evaluation       MIN06040
c           subroutine  fcn.                                             MIN06050
c                                                                        MIN06060
c        w  is a scratch vector.                                         MIN06070
c                                                                        MIN06080
c     on output                                                          MIN06090
c                                                                        MIN06100
c        x  contains the estimate of the minimum                         MIN06110
c           (xm(1),xm(2),...,xm(n))                                      MIN06120
c                                                                        MIN06130
```

```
c     ierr  is a parameter set to a negative integer if the user        MIN06140
c           wishes to force an exit from  search. otherwise it is       MIN06150
c           unaltered.                                                  MIN06160
c                                                                       MIN06170
c        g  contains the components of the gradient at  x.              MIN06180
c                                                                       MIN06190
c        f  contains the function value  f(x).                          MIN06200
c                                                                       MIN06210
c        alpha  is the final step sca                                   MIN06220
c                                                                       MIN06230
c        dirdev  is the directional derivative at  x.                   MIN06240
c                                                                       MIN06250
c        nfcall  is the number of calls to the function evaluation      MIN06260
c           subroutine  funct.                                          MIN06270
c                                                                       MIN06280
c                                                                       MIN06290
c     initialize the following parameters and indicators                MIN06300
c                                                                       MIN06310
c       tot       -  the sum of the extrapolation steps                 MIN06320
c       cdirev    -  the current directional derivative                 MIN06330
c       pdirev    -  the previous directional derivative                MIN06340
c       ierr      -  the error indicator                                MIN06350
c                                                                       MIN06360
      tot = 0.0d0                                                       MIN06370
      cdirev = dirdev                                                   MIN06380
      pdirev = dirdev                                                   MIN06390
c                                                                       MIN06400
c     test whether  alpha  is too small                                 MIN06410
c                                                                       MIN06420
  105 if (alpha .le. 1.0d-20) go to 150                                 MIN06430
c                                                                       MIN06440
c     begin the linear search by incrementing the solution vector  x    MIN06450
c       and calculating the function and gradient at the incremented  x.MIN06460
c                                                                       MIN06470
      do 108 i = 1, n                                                   MIN06480
         w(i) = x(i)                                                    MIN06490
         x(i) = x(i) + alpha * s(i)                                     MIN06500
  108 continue                                                          MIN06510
c                                                                       MIN06520
c     ****************************************************************  MIN06530
      call funct(n,x,ftest,g)                                           MIN06540
c     ****************************************************************  MIN06550
c                                                                       MIN06560
      nfcall = nfcall + 1                                               MIN06570
      if (maxf.lt.nfcall) go to 160                                     MIN06580
      if (ierr .lt. 0) go to 150                                        MIN06590
c                                                                       MIN06600
c     compute the directional derivative  dirdev  at  x + alpha * s     MIN06610
c                                                                       MIN06620
      dirdev = 0.0d0                                                    MIN06630
c                                                                       MIN06640
      do 110 i = 1, n                                                   MIN06650
         dirdev = dirdev + g(i) * s(i)                                  MIN06660
  110 continue                                                          MIN06670
c                                                                       MIN06680
c     test whether  f(x + alpha * s)  is less than  f(x).               MIN06690
```

```
c                                                                       MIN06700
      if (ftest .ge. f) go to 120                                       MIN06710
c                                                                       MIN06720
c     if (dirdev / pdirev) is less than the search precision - sprec,   MIN06730
c       alpha  is accepted. otherwise  alpha  is modified               MIN06740
c                                                                       MIN06750
      if (abs(dirdev / pdirev) .le. sprec) go to 140                    MIN06760
c                                                                       MIN06770
c     alpha  is modified. test whether  alpha  is to be revised by      MIN06780
c       extrapolation or interpolation                                  MIN06790
c                                                                       MIN06800
      if (dirdev .gt. 0.0d0) go to 120                                  MIN06810
c                                                                       MIN06820
c     alpha  is revised using an extrapolation formula and a new step   MIN06830
c       is taken if the sum of the steps already made is not too        MIN06840
c       the input parameter  extbnd  limits the multiplicative change   MIN06850
c       in  alpha                                                       MIN06860
c                                                                       MIN06870
      tot = tot + alpha                                                 MIN06880
      if (tot .gt. 1.0d10) go to 145                                    MIN06890
      temp = extbnd                                                     MIN06900
      if (cdirev .lt. dirdev) temp = dirdev / (cdirev - dirdev)         MIN06910
      if (temp .gt. extbnd) temp = extbnd                               MIN06920
      f = ftest                                                         MIN06930
      cdirev = dirdev                                                   MIN06940
      alpha = alpha * temp                                              MIN06950
      go to 105                                                         MIN06960
c                                                                       MIN06970
c     x  is reset to the current estimate, alpha  is revised using the  MIN06980
c       cubic interpolation formula and a new step is taken if the      MIN06990
c       convergence criteria have not been satisfied.                   MIN07000
c                                                                       MIN07010
  120 do 130 i = 1, n                                                   MIN07020
        x(i) = w(i)                                                     MIN07030
  130 continue                                                          MIN07040
c                                                                       MIN07050
      temp = 3.0d0 * (f - ftest) / alpha + dirdev + cdirev              MIN07060
      wt = abs(temp)                                                    MIN07070
      if(wt.lt.abs(dirdev)) wt=abs(dirdev)                              MIN07080
      if(wt.lt.abs(cdirev)) wt=abs(cdirev)                              MIN07090
      ww = temp / wt                                                    MIN07100
      ww = ww * ww - cdirev / wt * dirdev / wt                          MIN07110
      if (ww .lt. 0.0d0) ww = 0.0d0                                     MIN07120
      ww = dsqrt(ww) * wt                                               MIN07130
      temp = 1.0d0 - (dirdev + ww - temp) / (2.0d0 * ww + dirdev -      MIN07140
     1       cdirev)                                                    MIN07150
      alpha = alpha * temp                                              MIN07160
      go to 105                                                         MIN07170
c                                                                       MIN07180
c     alpha is accepted                                                 MIN07190
c                                                                       MIN07200
  140 f = ftest                                                         MIN07210
  145 alpha = tot + alpha                                               MIN07220
  150 return                                                            MIN07230
c                                                                       MIN07240
  160 do 170 i = 1, n                                                   MIN07250
```

```
          x(i) = w(i)                                                    MIN07260
  170 continue                                                           MIN07270
      return                                                             MIN07280
c                                                                        MIN07290
      end                                                                MIN07300
                                                                         MIN07310
      SUBROUTINE SUMDATA(NVAR,X,OBJF,G)                                  MIN07320
C     FILE TO GENERATE FN(T) FOR FISCHER-TROPSCH MODELLING AND FIT DATA  MIN07330
C     N IS CARBON NUMBER,NO IS THE NO OF DATA POINTS                     MIN07340
C     TAU AND TAUB ARE GUESSES                                           MIN07350
                                                                         MIN07360
      IMPLICIT REAL*8(A-H,O-Z)                                           MIN07370
      COMMON /EXPTDATAR/T,FEXP                                           MIN07380
      COMMON /EXPTDATAI/NO                                               MIN07390
      COMMON /CARBON/N                                                   MIN07400
      DIMENSION T(100),FEXP(100),X(NVAR),G(NVAR),Y(2)                    MIN07410
C*********************************************************************** MIN07420
C     OBJF IS SUM OF(F(EXPT)-F(T))**2;DOFTAU IS D/DTAU OF OBJF;          MIN07430
C     DOFTAUB IS D/DTAUB OF OBJF                                         MIN07440
C*********************************************************************** MIN07450
                                                                         MIN07460
      OBJF=0.0D0                                                         MIN07470
      DOFTAU=0.0D0                                                       MIN07480
      DOFTAUB=0.0D0                                                      MIN07490
      DTAU=0.0D0                                                         MIN07500
      DTAUB=0.0D0                                                        MIN07510
      DELT=1.0D-5                                                        MIN07520
                                                                         MIN07530
      DO 2 I=1,NO                                                        MIN07540
         Y(1)=X(1)                                                       MIN07550
         Y(2)=X(2)                                                       MIN07560
         TIME=T(I)                                                       MIN07570
         CALL FUNC(TIME,FN,DELTAU,DELTAUB,X,NVAR)                        MIN07580
         OBJF=OBJF+(FEXP(I)-FN)**2                                       MIN07590
         DOFTAU=DOFTAU+(-2.0D0*(FEXP(I)-FN)*DELTAU)                      MIN07600
         DOFTAUB=DOFTAUB+(-2.00*(FEXP(I)-FN)*DELTAUB)                    MIN07610
         Y(1)=X(1)*(1.0 + DELT/2. )                                      MIN07620
         CALL FUNC(TIME,FNT,DELTA,DELTAB,Y,NVAR)                         MIN07630
         Y(1)=X(1)*(1.0 - DELT/2. )                                      MIN07640
         CALL FUNC(TIME,FNT1,DELTA,DELTAB,Y,NVAR)                        MIN07650
         DTAU=DTAU+(-2.0D0*(FEXP(I)-FN)*(FNT-FNT1)/(X(1)*DELT))          MIN07660
                                                                         MIN07670
         Y(1)=X(1)                                                       MIN07680
         Y(2)=X(2)*(1.0 + DELT/2.0D0)                                    MIN07690
         CALL FUNC(TIME,FNTB,DELTA,DELTAB,Y,NVAR)                        MIN07700
         Y(2)=X(2)*(1.0 - DELT/2.0D0)                                    MIN07710
         CALL FUNC(TIME,FNTB1,DELTA,DELTAB,Y,NVAR)                       MIN07720
         DTAUB=DTAUB+(-2.0D0*(FEXP(I)-FN)*(FNTB-FNTB1)/(X(2)*DELT))      MIN07730
                                                                         MIN07740
 2    CONTINUE                                                           MIN07750
                                                                         MIN07760
      WRITE(6,*)'O.F',OBJF,DTAU,DTAUB                                    MIN07770
      WRITE(6,*)'TAU=',X(1),'TAUB',X(2)                                  MIN07780
c     G(1)=DOFTAU                                                        MIN07790
c     G(2)=DOFTAUB                                                       MIN07800
      G(1)=DTAU                                                          MIN07810
```

```
      G(2)=DTAUB                                                        MIN07820
                                                                        MIN07830
      RETURN                                                            MIN07840
      END                                                               MIN07850
                                                                        MIN07860
      SUBROUTINE FACT(M,J)                                              MIN07870
      J=1                                                               MIN07880
      IF(M.EQ.0)GOTO 4                                                  MIN07890
      DO 1 I=1,M                                                        MIN07900
      J=J*I                                                             MIN07910
 1    CONTINUE                                                          MIN07920
4     RETURN                                                            MIN07930
      END                                                               MIN07940
                                                                        MIN07950
C***********************************************************************MIN07960
C     FUNC EVALUATES FN(TIME),AND PARTIAL DERIVATIVES WRT TAU AND TAUB  MIN07970
C         DELTAU AND  DELTAUB                                           MIN07980
C***********************************************************************MIN07990
      SUBROUTINE FUNC(TIME,FN,DELTAU,DELTAUB,X,NVAR)                    MIN08000
      IMPLICIT REAL*8(A-H,O-Z)                                          MIN08010
      COMMON /CARBON/N                                                  MIN08020
      DIMENSION X(NVAR)                                                 MIN08030
      TAU=X(1)                                                          MIN08040
      TAUB=X(2)                                                         MIN08050
      A1=TAUB/(TAU-TAUB)                                                MIN08060
      AK1=0.0D0                                                         MIN08070
      AK2=0.0D0                                                         MIN08080
      DO 1 I=1,N                                                        MIN08090
      AK1=AK1+(A1**I)*((-1)**(I+1))                                     MIN08100
 1    CONTINUE                                                          MIN08110
      AK2=AK2+((-A1)-1.)                                                MIN08120
      DO 2 I=2,N                                                        MIN08130
C     WRITE(22,*)TIME,NI                                                MIN08140
      DO 3 IR1=1,I-1                                                    MIN08150
      IR=IR1-1                                                          MIN08160
      CALL FACT(I-IR-1,IRR)                                             MIN08170
C     WRITE(22,*)TIME,IRR                                               MIN08180
      IF(IRR.LE.0)GOTO 200                                              MIN08190
      AK2=AK2+(1.000/(TAU**I))*(TAU**(IR+1)*TIME**(I-IR-1)*(-1+(-A1)**(IMIN08200
     1R+1)))/IRR)                                                       MIN08210
 3    CONTINUE                                                          MIN08220
      AK2=AK2+(1.000)*((-A1)**(I)-1.)                                   MIN08230
 2    CONTINUE                                                          MIN08240
      T1=TIME/TAU                                                       MIN08250
      T2=TIME/TAUB                                                      MIN08260
      IF(T1.LT.75.D0.AND.T2.LT.75.D0)GO TO 20                           MIN08270
      FN=1.0D0                                                          MIN08280
      IF(T1.LT.75.D0)FN=1.0D0+AK2*DEXP(-TIME/TAU)/N                     MIN08290
      IF(T2.LT.75.D0)FN=1.0D0+AK1/N*DEXP(-TIME/TAUB)                    MIN08300
      GO TO 21                                                          MIN08310
20    FN=1.0+(AK1/N)*DEXP(-TIME/TAUB)+AK2*DEXP(-TIME/TAU)/N             MIN08320
21    AL1=0.0D0                                                         MIN08330
      AL2=AK2                                                           MIN08340
      AL3=0.0D0                                                         MIN08350
      AL4=0.0D0                                                         MIN08360
      DO 4 I=1,N                                                        MIN08370
```

```
      AL1=AL1+(-TAUB)**I*I/((TAU-TAUB)**(I+1))                          MIN08380
    4 CONTINUE                                                          MIN08390
      AL3=AL3+1.0*((-A1)-1.0D0)/TAU                                     MIN08400
      AL4=AL4+1.0D0/TAU*((-A1-1.0D0)-TAU*(-A1)/(TAU-TAUB))              MIN08410
      DO 5 I=2,N                                                        MIN08420
      DO 6 IR1=1,I-1                                                    MIN08430
      IR=IR1-1                                                          MIN08440
      CALL FACT(I-IR-1,IRR)                                             MIN08450
      IF(IRR.LE.0)GO TO 200                                             MIN08460
      AL3=AL3+1.00/TAU**(I+1)*I*(TAU)**(IR+1)*TIME**(I-IR-1)/IRR*(-1+(-AMIN08470
     11)**(IR+1))                                                       MIN08480
      AL4=AL4+1.0/TAU**I*TIME**(I-IR-1)/IRR*((IR+1)*TAU**IR*(-1+(-A1)**(MIN08490
     1IR+1))-(TAU**(IR+1)*((IR+1)*(-A1)**(IR+1)/(TAU-TAUB))))           MIN08500
    6 CONTINUE                                                          MIN08510
      AL3=AL3+I/TAU*((-A1)**I-1.0D0)                                    MIN08520
      AL4=AL4+I/TAU*((-A1)**I-1.0D0-TAU*(-A1)**I/(TAU-TAUB))            MIN08530
    5 CONTINUE                                                          MIN08540
      IF(T1.LT.75.D0.AND.T2.LT.75.D0)GO TO 22                           MIN08550
      DELTAU=0.0D0                                                      MIN08560
      IF(T1.LT.75.D0)DELTAU=DEXP(-T1)/N*(TIME/TAU**2*AL2-AL3+AL4)       MIN08570
      IF(T2.LT.75.D0)DELTAU=AL1/N*DEXP(-TIME/TAUB)                      MIN08580
      GO TO 23                                                          MIN08590
   22 DELTAU=AL1/N*DEXP(-TIME/TAUB)+DEXP(-TIME/TAU)/N*(TIME/TAU**2*AL2-AMIN08600
     1L3+AL4)                                                           MIN08610
C     WRITE(6,*)'TIME',TIME,'DELTAU',DELTAU,X(1),X(2)                   MIN08620
   23 AM2=0.0                                                           MIN08630
      AM3=0.0                                                           MIN08640
      DO 7 I=1,N                                                        MIN08650
    7 AM2=AM2+(-1)**(I+1)*I*TAU*TAUB**(I-1)/(TAU-TAUB)**(I+1)           MIN08660
      AM3=AM3-TAU/(TAU-TAUB)**2                                         MIN08670
      DO 8 I=2,N                                                        MIN08680
      DO 9 IR1=1,I-1                                                    MIN08690
      IR=IR1-1                                                          MIN08700
      AM3=AM3-1./TAU**I*(TAU**(IR+2)*TIME**(I-IR-1)*(IR+1)*(-TAUB)**IR/IMIN08710
     1RR/(TAU-TAUB)**(IR+2))                                            MIN08720
    9 CONTINUE                                                          MIN08730
      AM3=AM3-TAU*I*(-A1)**(I-1)/(TAU-TAUB)**2                          MIN08740
    8 CONTINUE                                                          MIN08750
      IF(T1.LT.75.D0.AND.T2.LT.75.D0)GO TO 24                           MIN08760
      DELTAUB=0.0D0                                                     MIN08770
      IF(T2.LT.75.D0)DELTAUB=EXP(-T2)/N*(TIME/TAUB**2*AK1+AM2)          MIN08780
      IF(T1.LT.75.D0)DELTAUB=AM3/N*DEXP(-T1)                            MIN08790
      RETURN                                                            MIN08800
   24 DELTAUB=EXP(-TIME/TAUB)/N*(TIME/TAUB**2*AK1+AM2)+EXP(-TIME/TAU)/N*MIN08810
     1AM3                                                               MIN08820
      RETURN                                                            MIN08830
  200 WRITE(6,201)                                                      MIN08840
  201 FORMAT(1X, 'FACTORIAL RETURNED NEGATIVE')                         MIN08850
      END                                                               MIN08860
                                                                        MIN08870
                                                                        MIN08880
```

END

DATE
FILMED
8/27/92