

Pertti Kärki

I/O-suhdesäätö alkylointiyksikössä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikan koulutusohjelma

Insinöörityö

24.11.2015

Tekijä Otsikko	Pertti Kärki I/O-suhdesäätö alkylointiyksikössä
Sivumäärä Aika	24 sivua + 4 liitettä 24.11.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Prosessiautomaatio
Ohjaaja Ohjaava opettaja	Kehityspäällikkö Marja Sarikkola Lehtori Jukka-Pekka Pirinen
<p>Insinööriyön aiheena oli tutkia Neste Oilin Porvoon jalostamon alkylointiyksikön I/O-suhdesäädön laskentaa ja selvittää laskennan sekä siihen liittyvien mittausten ja analysaattoreiden luotettavuutta. Oli havaittu, että I/O-suhdesäätö rajoittaa tuotantoa aiempaa enemmän, eikä laskentaa pidetty luotettavana. Yhtenä insinööriyön osana oli tuottaa oman henkilökunnan käyttöön laskentataulukko, jolla reaaliaikaisen isobutaani/olefiini-suhdelaskennan oikeellisuus kyettäisiin helposti tarkastamaan. Tuloksena saatu I/O-suhdelaskentataulukko nopeuttaa huomattavasti I/O-suhteen oikeellisuuden tarkastusta ongelmatilanteissa.</p> <p>Työn alkuvaiheessa tarkastettiin kaikki I/O-suhdesäätöön liittyvät mittaukset ja analysaattorit sekä suoritettiin tarvittavat kalibroinnit. Havaittiin, että mittausten impulssiputkiin saattaa kertyä putkia tukkivia partikkeleita, mikä voi aiheuttaa huomattavia vääristymiä mittauksissa. Virtausmittausten sekä analysaattoreiden säännöllisellä huollolla voidaan ennaltaehkäistä likaantumista johtuvia prosessihäiriöitä sekä turhia tuotantotappioita. Mittausten ja analysaattoreiden huollot sekä yksikön koeajo mahdollistivat syöttötason noston 30:stä 32,5 t/h. Syöttötason nosto paransi yksikön kannattavuutta noin 300 k€ kuukaudessa.</p>	
Avainsanat	Alkylointi, I/O-suhde

Author Title	Pertti Kärki I/O-ratio control in alkylation unit
Number of Pages Date	24 pages + 4 appendices 24 November 2015
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Process Automation
Instructor(s)	Marja Sarikkola, Development Manager Jukka-Pekka Pirinen, Lecturer
<p>The purpose of this thesis was to study I/O ratio of alkylation unit at Neste Oyj oil refinery in Porvoo and to examine the reliability of the I/O ratio calculation, meters and analyzers. Another aim of the thesis was to provide Neste personnel a calculation table which would allow the staff to check the validity of real time I/O ratio calculation. It was found that the I/O ratio control limited the alkylate production more than before and the staff did not trust the calculation behind the ratio.</p> <p>At the beginning of the thesis, all meters and analyzers associated with the calculation were checked and calibrated. It was detected that there might be mass pipeline blocking particles in the impulse lines of the meters, which can cause a remarkable distortion in the meters. With regular maintenance of the flow meters and analyzers, it is possible to prevent process failures and production losses caused by dirt from occurring. The maintenance of the meters and analyzers and the unit test drive enabled the unit feed rate to be lifted from 30 t/h to 32.5 t/h. The feed rate lift increased the unit's profitability by approximately 300 k€ per month.</p>	
Keywords	Alkylation, I/O-ratio

Sisällys

Lyhenteet

1	Johdanto	1
2	Porvoon jalostamo	2
3	Alkylointi	3
3.1	Alkylointiyksikkö	3
3.2	Alkylointiprosessi	3
4	I/O-suhdesäätö	5
4.1	Teoria	5
4.2	Mittaukset	5
4.3	Analysaattorit	7
4.4	Laskenta	8
4.5	Säädöt	10
5	Koeajot	11
5.1	Virtausmittausten kalibrointi	11
5.2	Analysaattoreiden luotettavuuden todentaminen	12
5.3	Suhdesäädön muutokset	17
6	Päätelmät	18
6.1	Virtausmittausten toiminta	18
6.2	Analysaattoreiden toiminta	18
6.3	Koeajot	19
6.4	Prosessihäiriöt	21
6.5	I/O-suhde-Excel	22
7	Tulosten yhteenveto	23
	Lähteet	24
	Liitteet	
	Liite 1. Analysaattoreiden pitoisuudet	
	Liite 2. Koeajosuunnitelma	
	Liite 3. Olefiinisyötön laboratorioanalyysi	
	Liite 4. TOP-järjestelmän I/O-laskurin koodi	

Lyhenteet

I/O	Isobutaani / Olefiini
IC4	Isobutaani
C5+	Pentaani sekä sitä pitkäketjuisemmat hiilivedyt
RON	Standardiolosuhteille ja –moottoreille saatu oktaaniluku
MON	Huipputehoa tiekäytössä simuloiva oktaaniluku
m-%	Massaprosentti
möl-%	Mooliprosentti
TOP	Jalostamonlaajuinen prosessinohjaus- ja informaatiojärjestelmä, josta saadaan asetusarvot osalle prosessiyksiköiden säätimistä
ALKY	Alkylointiyksikkö
NKJT	Nestekaasujen jakotislausyksikkö
MTBE	Metyylitertiääributyylieetteriyksikkö
BDY	Butadieenien hydrausyksikkö
OKSY	Oksygenaattien poistoyksikkö
PTM	Pieni tekninen muutos, jalostamalla käytössä oleva muutosten riskitarkasteluohjelma

1 Johdanto

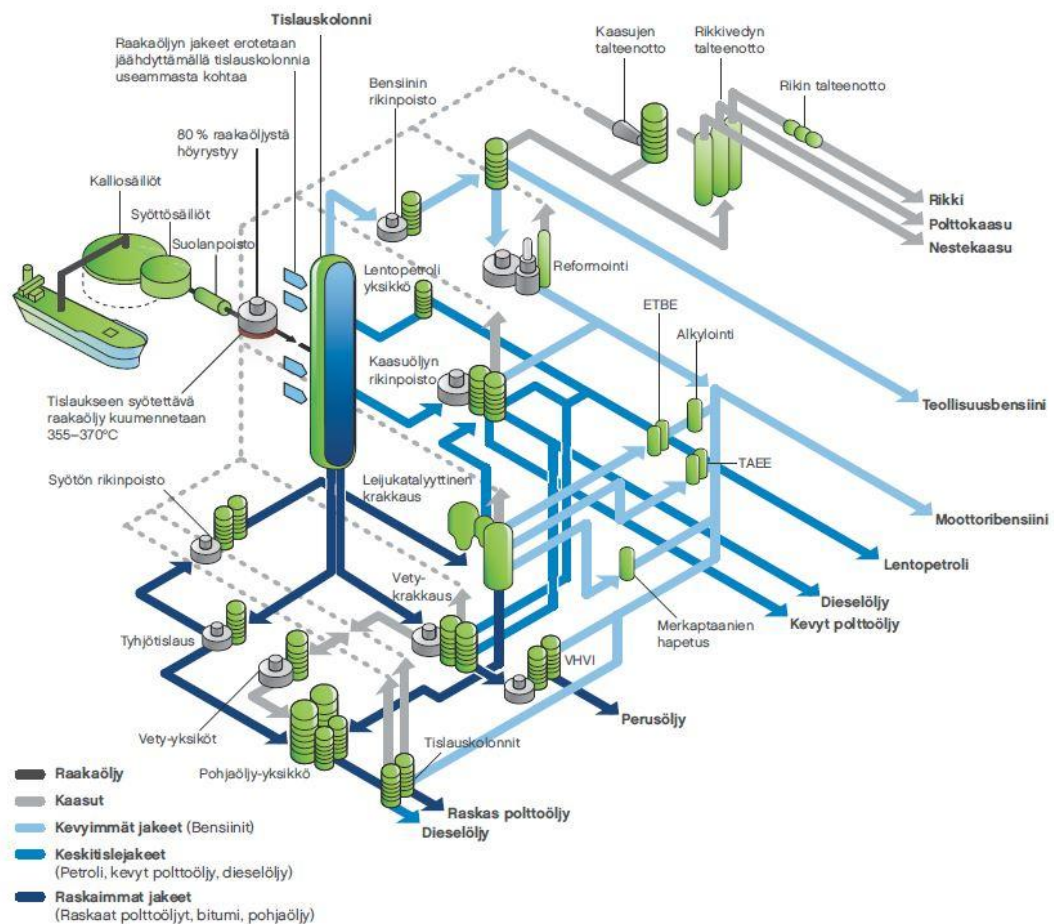
Nesteen Porvoon jalostamo on kymmenistä eri prosessiyksiköistä koostuva kompleksinen jalostamo, jonka tavoitteena on tuottaa pitkälle kehitettyjä puhtaampia polttoainevaihtoehtoja. Yksi arvokkaimmista bensiinikomponenteista on alkylaatti, jota valmistetaan alkylointiyksikössä fluorivetyhapon avulla.

Alkylaatin tuotannossa oli huomattu normaalista poikkeavia kapasiteettirajoitteita, joille ei löytynyt selitystä. Alkylointiyksikön yksi tärkeimmistä mittareista on yksikön I/O-suhde, jonka perusteella säädetään yksikön olefinisyötön määrää. Syöttötasolla on suora vaikutus yksikön tuotesaantoon.

Työn teoriaosassa käydään läpi I/O-suhteen laskentaa sekä siihen liittyviä prosessi-muuttujia. Tavoitteena on rakentaa Excel-laskuri, joka vastaa jalostamon käytössä olevan informatiivisen TOP-järjestelmän reaaliaikaista laskentaa sekä tutkia, antaako laskenta oikean lopputuloksen. Käytännön osuudessa tarkoitus on selvittää I/O-suhdelaskennan toimivuutta koeajoilla sekä haastatella tuotannon operaattoreita suhdettäön liittyvistä yksikön pullonkauloista.

2 Porvoon jalostamo

Porvoon jalostamon toiminta on aloitettu vuonna 1965. Jalostamo on yksi Euroopan nykyaikaisimpia ja monipuolisimpia jalostamoita. Raakaöljyn tislaukaskapasiteetti on noin 200 000 tynnyriä päivässä ja tuotanto noin 12,5 miljoonaa tonnia vuodessa. Jalostamo muodostuu neljästä eri tuotantolinjasta ja yli 40 prosessiyksiköstä. Valmistuksessa on yli 150 tuotetta tai tuotekomponenttia. Tuotannosta yli 60 % menee kotimaan markkinoille ja loput vientiin lähinnä Pohjoismaihin, Keski-Eurooppaan sekä Pohjois-Amerikkaan. Suurin osa raaka-aineista ja tuotteista kuljetetaan meriteitse. Jalostamon satamassa käy vuosittain 1200 -1400 laivaa. Nesteen noin 5000 työntekijästä 1900 työskentelee Porvoon jalostamolla.



Kuva 1. Porvoon jalostamo

3 Alkylointi

3.1 Alkylointiyksikkö

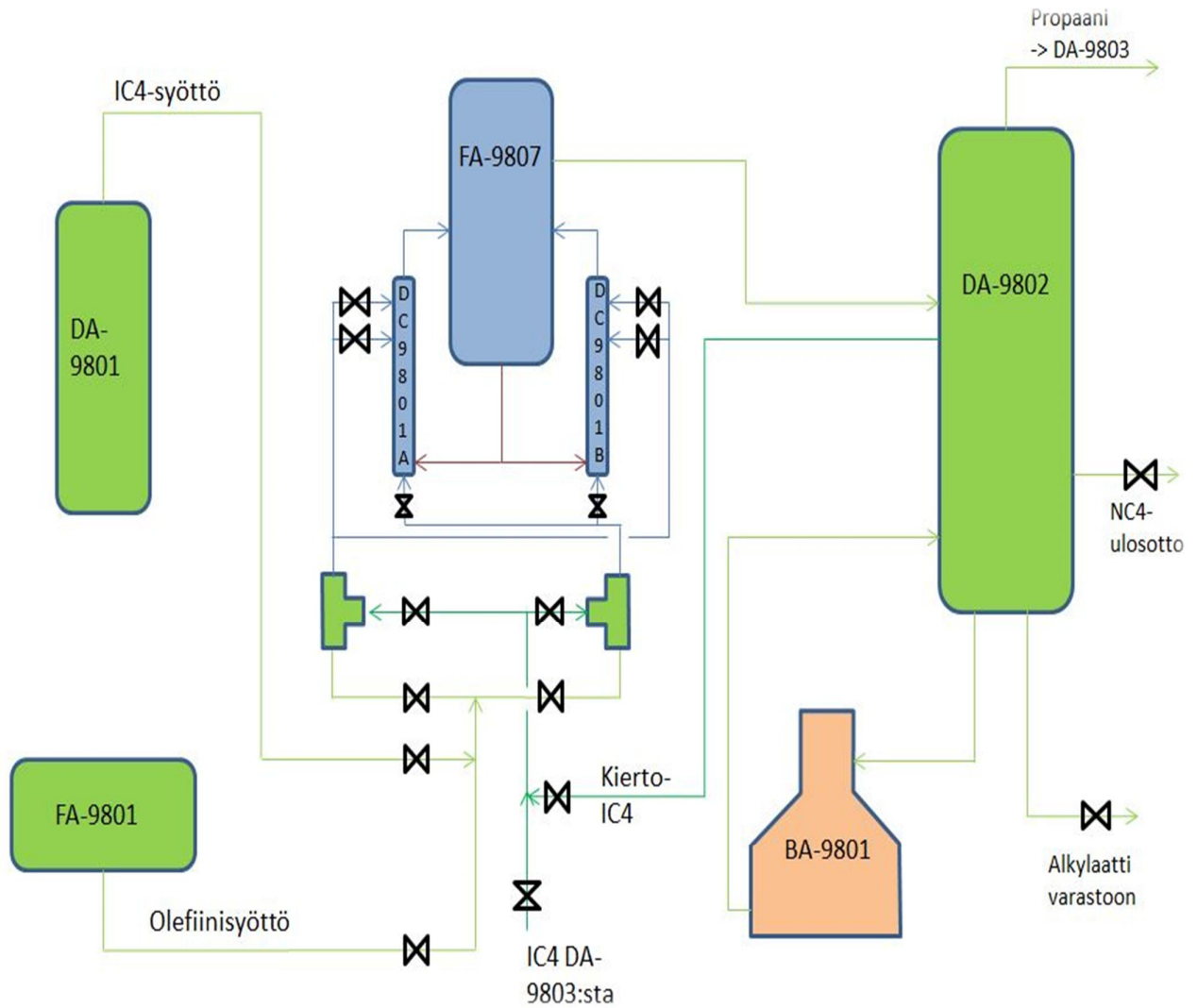
Alkylointiyksikön tehtävä on tuottaa nestekaasuista korkealaatuista bensiinikomponenttia, alkylaattia. Yksikkö on suunniteltu 40,9 t/h kokonaissyötölle, josta olefiinisyöttöä on 29,9 t/h. Suunniteltu alkylaatin tuotanto on 33,8 t/h. Tämän tavoitteen saavuttamiseksi on otettu käyttöön Phillips Petroleumin patentoima, Split Olefin Feed Technology, jaetun olefiinisyötön tekniikka. Tämänhetkisillä syöttöainekoostumuksilla on asetettu olefiinisyötölle yläraja 32,5 t/h. (Alkylointiyksikön yleisohje v.10.)

Jaetun olefiinisyötön teoria perustuu tehokkaamman isobutaanin hyödyntämiseen alentamalla kokonais-I/O-suhdetta. Tärkein etu on kasvattaa prosessin operoitavuutta optimoimalla isobutaanin ja olefiinien kontakti reaktoriputkessa. Olefiinisyöttö, kiertoisobutaani ja tuoreisobutaani sekoitetaan tietyssä suhteessa keskenään ja syötetään reaktoriputkissa eri korkeuksilla olevien syöttösuuttimien läpi.

3.2 Alkylointiprosessi

Alkylointiprosessissa buteeni yhdistetään isobutaanin kanssa. Osa tarvittavasta isobutaanista on olefiinisyötön mukana, loppuosa saadaan isobutaanitisleenä DA-9801:stä. Olefiini- ja isobutaanimolekyylit reagoivat keskenään fluorivetyhapon toimiessa katalyyttinä. Fluorivetyhappo osallistuu reaktioon, mutta vapautuu reaktion loputtua. Syntynyttä alkylaattia käytetään korkeaktaanisten bensiinien komponenttina. Tuotteen, alkylaatin, RON() on 96, MON() 94 ja suunniteltu tuotanto 33,8 t/h.

Alkylointireaktio on lämpöä vapauttava eksotermiäinen reaktio. Alkylaatti, propaani, butaani, happo ja isobutaani pysyvät sekoittuneena virratessaan ylös reaktoriputkissa DC-9801A/B. Tiheysero lämpimän reaktoriputken happo/hiilivedyn ja viileämmän hapon kesken aikaansaavat termosifooni-virtauksen ja pakottavat hapon selkeyttimeen FA-9807. Näiden seos pysyy sekoittuneessa muodossa johtuen virtausnopeudesta reaktoriputkessa. Hiilivetyjen viipymäaika reaktoriputkissa on noin 20 s. (Alkylointiyksikön yleisohje v.10.)



Kuva 2. Alkylointiprosessi

Olefiinisyytön mukana on pieniä pitoisuuksia vettä sekä prosessille haitallisia yhdisteitä, jotka muodostavat pitkäketjuisia hiilivetykomponentteja, ASO:a. Vesi ja happoon liukeneva öljy, ASO, laimentavat happoa. Nämä epäpuhtaudet on poistettava haposta sen riittävän puhtauden säilyttämiseksi. Osa haposta menetetään hapon regeneroinnissa, osa poistuu yksiköstä orgaanisina fluorideina propaanin, n-butaanin ja alkylaatin mukana. (Alkylointiyksikön yleisohje v.10.)

4 I/O-suhdesäätö

4.1 Teoria

I/O-suhdesäädön tarkoituksena on optimoida alkylointiyksikön reaktoriputkiin syötettävän isobutaanin ja olefiinien tilavuuksien suhde.

Porvoossa alkyloinnin kokonais-I/O-suhteelle on asetettu alarajaksi $8 \text{ m}^3 / \text{m}^3$. On havaittu, että alle 7 I/O-suhteella prosessi alkaa tuottamaan suuria määriä ASO:a sekä polymeerejä, jotka tukkivat putkistoja sekä pilaavat alkylaatin laadun. Vastaavasti korkeammalla I/O-suhteella saadaan hyvää alkylaattia ja vähän ASO:a, mutta yksikön olefiinisyöttöä joudutaan rajoittamaan, mikä puolestaan vähentää alkylaatin tuotantoa. (Alkylointiyksikön yleisohje v.10.)

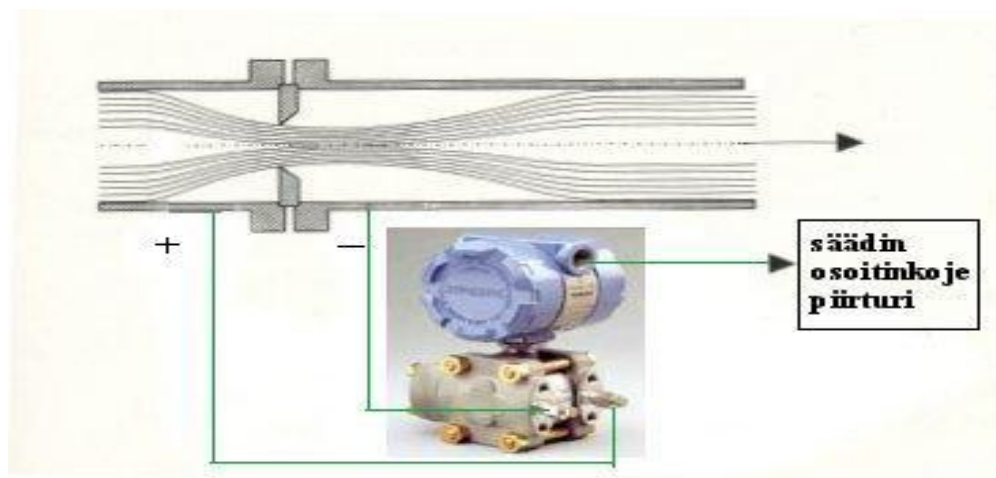
4.2 Mittaukset

I/O-suhdesäädön laskennassa käytetään virtausmittauksia, jotka kaikki perustuvat paine-eromittauksiin. Virtaava aine (hiilivety) on jokaisessa mittauskohdassa nestemäisessä olomuodossa.

Paine-eroon perustuvassa virtausmittauksessa suoralle putkiosuudelle asennetaan mittalaippa eli kuristuslaippa, jonka molemmilta puolilta mitataan paine. Suoran putkiosuuden suositellut vähimmäispituudet ennen ja jälkeen mittalaipan:

- ennen laippaa 10-20 x putken halkaisija
- laipan jälkeen 4-7 x putken halkaisija.

Mittalaipan yli olevan paine-eron avulla saadaan laskettua putken sisällä virtaavan aineen massavirtaus. (Sivonen 1999.)



Kuva 3. Paine-eroon perustuva virtausmittaus

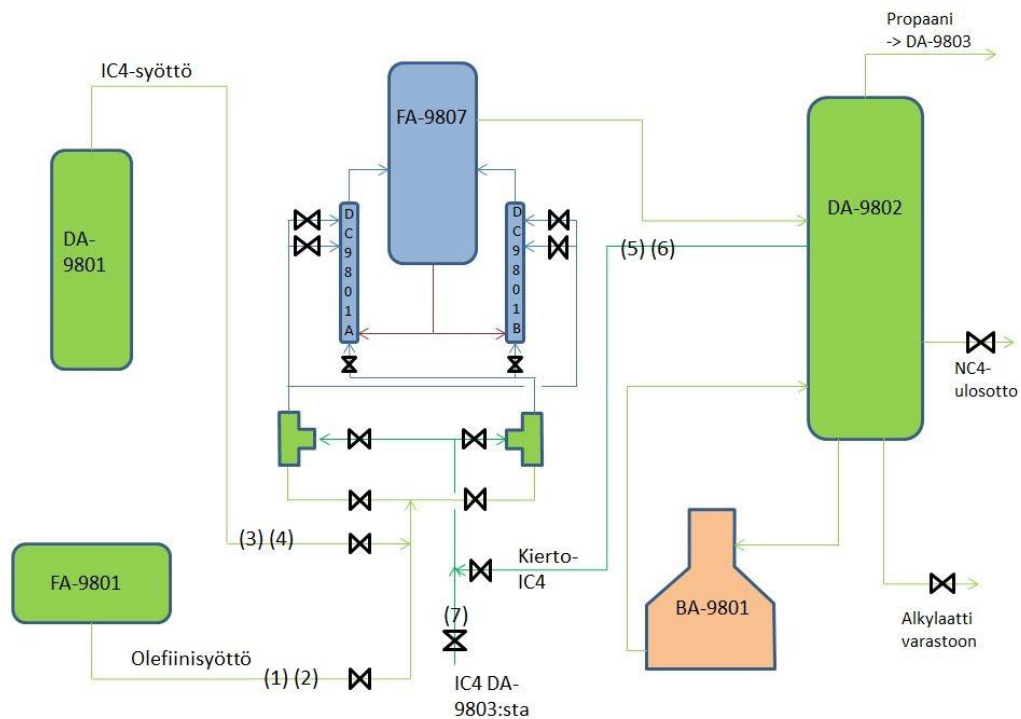
4.4 Laskenta

Alkyloinnin I/O-suhdelaskenta on toteutettu osaksi TOP-järjestelmää. TOP-järjestelmä on informatiivinen prosessiyksiköiden seurantajärjestelmä, joka ei laske I/O-suhteen osalta säätimille asetusrvoja, vaan antaa ainoastaan suhdeluvun I/O, jonka perusteella prosessia säädetään Metson automaatiojärjestelmässä. Laskuri on 48 A4-sivun mittainen tekstitiedosto, joka on toteutettu FORTRAN -ohjelmointikielellä. (Liite 3).

Laskenta ottaa huomioon seuraavat muuttujat:

Kokonais-I/O-suhde

- FCA9812 Olefiinisyöttö (1) +AI9820_1 Olefiinisyötön analysaattori (2)
- FCA9809 Isobutaanisyöttö (3) +AI9810_1 Isobutaanisyötön analysaattori (4)
- FI9820 Isobutaanikierto (5) +AI9810_2 Isobutaanikierron analysaattori (6)
- FC9843 DA-9803:n pohja (7)



Kuva 5. Virtausmittausten ja analysaattoreiden sijainnit.

Laskentaohjelmaan on syötetty prosessivirroissa mukana olevien eri hiilivetyjakeiden tiheydet ja moolimassat. Virtausmittausten (t/h), analysaattoritulosten (mol-%) sekä valmiiksi syötettyjen vakioarvojen avulla ohjelma laskee reaktoriputkiin syötettävän puhtaan olefiinijakeen ja isobutaanijakeen tilavuudet, jolloin saadaan laskettua isobutaani / olefiini -suhde m^3 / m^3 .

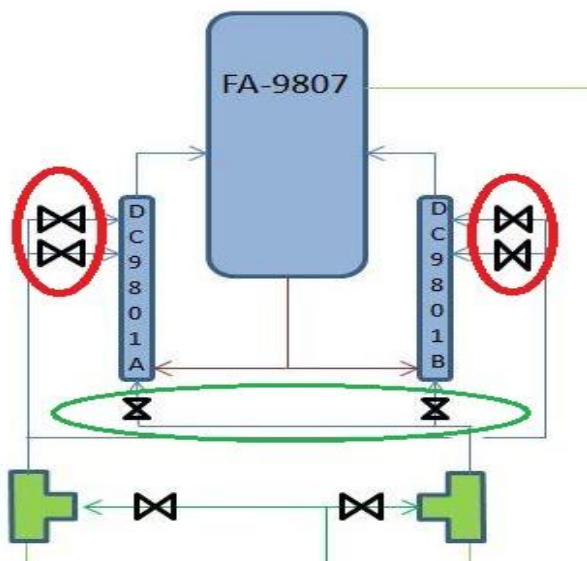
DA-9803:n pohjavirtaukselle ei ole analysaattoria. Virtaus on tonnimäärältään pieni kokonaisuuden kannalta ja sisältää keskimäärin yli 80 mol-% isobutaania. Laskentaan on kyseisen virtauksen pitoisuudeksi asetettu 85 m-%, mikä vastaa noin 82 mol-%.

Ylä- ja alasuuttimien I/O-suhde

Ylä- ja alasuuttimien I/O-suhteen laskemiseen tarvitaan kokonais-I/O-suhteen laskennassa käytetyt virtaukset ja analysaattoritulokset sekä

- FC9873 Tuoresyöttö yläsuuttimille
- FCA9881 Kiertosyöttö yläsuuttimille.

I/O-suhdelaskurista tehty Excel-taulukko kohdassa 6.5. Laskentakaavat saa auki Excellissä, joka on luovutettu tuotannon käyttöön.



Kuva 6. Alasuuttimet 2 kpl ympyröitynä vihreällä ja yläsuuttimet 4 kpl punaisella.

4.5 Säädot

ALKY:n olefiinisyöttö FCA9812 pyritään maksimoimaan tasolle 32,5 t/h kuitenkin niin, ettei I/O-suhde laske alle $8,0 \text{ m}^3 / \text{m}^3$. Olefiinisyötön mukana saadaan sekä olefiineja, isobutaania että normaalibutaania. Syötön normaali olefiinipitoisuus on noin 54 mol-%. Olefiinipitoisuuteen vaikuttavat MTBE-yksikön U24 lisäsyötön laatu sekä Borealikselta saatava raffinaatti. Raffinaattia ei voitu ottaa ALKY:n syöttöön ennen koeajoja, koska sen laatu olisi heikentänyt I/O-suhdetta, joka oli valmiiksi alarajalla 8,0.

Syöttövirtauksen lisäksi suhdetta voidaan muuttaa säätämällä isobutaanikierron FI9820 pitoisuuksia. IC4-kierron pitoisuuksia voidaan säätää lisäämällä tai vähentämällä propanin FC9849 tai n-butaanin FC9852 ulosottoa.

ALKY:n käytännön rajoitteista johtuen ylä- ja alasuuttimien I/O-suhdetta ei ole mahdollista muuttaa kovinkaan paljon. Alasuuttimien I/O-suhde on normaalisti $10\text{-}11 \text{ m}^3 / \text{m}^3$. Yläsuuttimien I/O-suhde tuoresyötössä jää tällöin tasolle $6\text{-}6,5 \text{ m}^3 / \text{m}^3$. Yläsuuttimien paikallinen I/O-suhde reaktoriputkessa nousee kuitenkin huomattavasti alasuuttimien kautta tulevan ylimääräisen isobutaanin myötä ollen yli $20 \text{ m}^3 / \text{m}^3$.

5 Koeajot

Ennen varsinaisia koeajoja selvitettiin I/O-suhdesäädössä mukana olevien virtausmittausten sekä analysaattoreiden luotettavuutta. Operaattoreilta saadun tiedon sekä yksikön massataselaskennan perusteella oli syytä olettaa, että osa mittauksista tai analysaattoreista näyttää väärin.

5.1 Virtausmittausten kalibrointi

Virtausmittausten impulssiputket puhallutettiin puhtaiksi ja samalla suoritettiin mittaus-ten nollakohdan tarkastus. Impulssiputkien puhallutuksella saadaan poistettua putkien seinämiin tarttunut tukkiva aines, mikäli sellaista on päässyt kertymään. Nollakohdan tarkastuksella saadaan poistettua mahdollinen anturista johtuva mittauspoikkeama.

Tuoreisobutaanin syöttövirtauksen FCA9809 tapauksessa impulssiputkiin oli kertynyt karstaa, mikä vääristi mittaustulosta. Kävi ilmi, että mittaus oli näyttänyt noin 2 t/h liian vähän. Kyseisellä mittauksella poikkeama oli 20 % todelliseen verrattuna.

Muissa virtausmittauksissa ei havaittu tuketta eikä mittavirheitä.

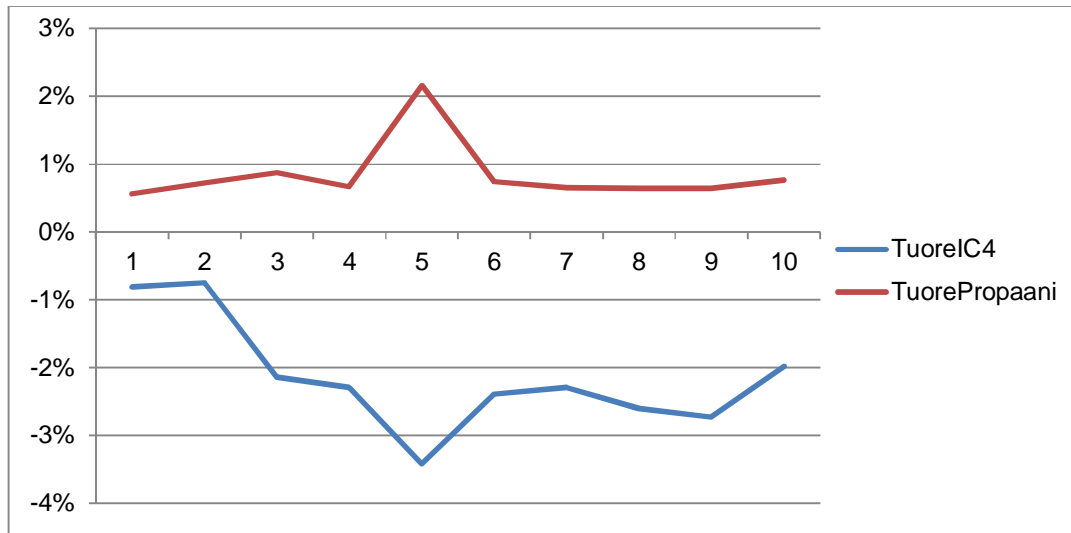
5.2 Analysaattoreiden luotettavuuden todentaminen

I/O-suhdesäädössä on mukana kolme analysaattoria: AI9810_1, AI9810_2 ja AI9820_1, jotka näyttävät niille määriteltyjen aineiden pitoisuudet prosessivirrasta mooliprosentteina. Lähtötilanteessa AI9810_2:n tulokset poikkesivat ajoittain yli 5 %-yksikköä jalostamon laboratorion tuloksista. Myös AI9820_1:llä havaittiin vastaavaa ongelmaa. Analysaattoriasentajien mukaan analysaattorin virhemarginaali on noin 2 %, joten prosessia jouduttiin säätämään laboratoriotulosten perusteella. AI9810_2:ta oli käyty puhdistamassa ja kalibroimassa huomattavasti normaalia useammin, eivätkä tulokset käyneet yksiin laboratoriotulosten kanssa.

Analysaattoreiden tuloksia tarkastellessa niistä löytyi yllättävä piirre: kaikkien aineiden yhteenlaskettu osuus prosessivirrasta poikkesi 100 %:sta. Jokaisella analysaattorilla on skaalaustoiminto (100 %), mutta se ei ollut päällä. Skaalauksen päälle oton jälkeen tuloksia voitiin luotettavasti vertailla laboratoriotuloksiin, koska laboratoriotulokset ovat aina skaalattuja antaen yhteenlaskettuna 100 %.

Seuraavaksi analysaattoreiden luotettavuutta selvitettiin viemällä laboratorioon analysoitavaksi vertailunäytteet viiden vuorokauden ajan 12 tunnin välein. Vertailussa on esitetty I/O-suhteen kannalta merkityksellisen aineiden analysaattorin ja laboratoriotuloksen ero.

AI9810_1

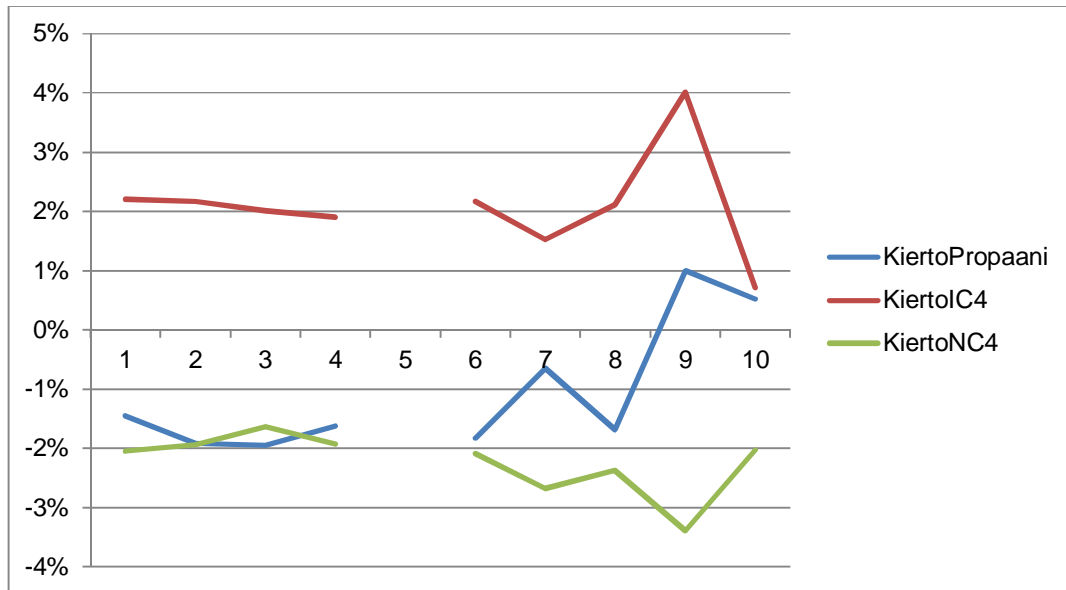


Kuva 7. AI9801_1-analysaattorin luotettavuusvertailu. Poikkeama ilmoitettu mol%-yksikköä analysaattori-laboratoriotulos

Tuoreisobutaanisyötön analysaattorin ja laboratoriotulosten välistä eroa tarkastellessa viidennen näytteen tulos ei ole vertailukelpoinen, koska sillä hetkellä prosessiin oltiin tekemässä muutosta. Analysaattori ottaa näytteen prosessivirrasta noin 25 minuutin välein, joten vertailukelpoisen näytteen saaminen edellyttää, että prosessissa ei tapahdu muutosta näytteenottohetkellä.

Tulosten perusteella analysaattori poikkeaa laboratoriotuloksesta tasaisesti, mutta poikkeama ei elä yli 2 %-yksikköä. IC4:n pitoisuus virrasta on normaalitilanteessa noin 90 mol-%.

AI9810_2



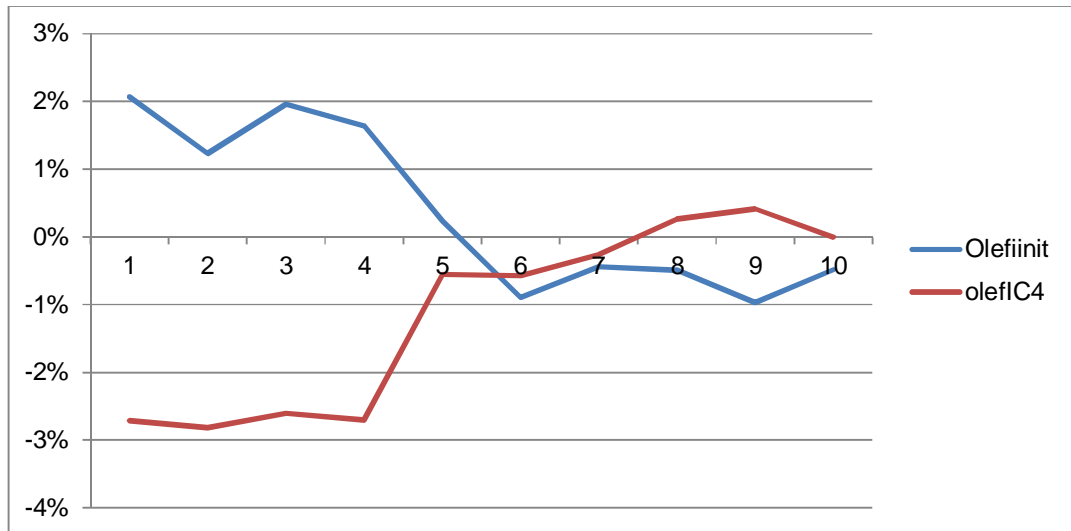
Kuva 8. AI9810_2-analysaattorin luotettavuusvertailu. Poikkeama ilmoitettu mol%-yksikköä analysaattori-laboratoriotulos.

Isobutaanikierron viides laboratorioanalyysi jätettiin viemättä prosessin muutoksen vuoksi. Toiseksi viimeisen näytteen ottohetkellä prosessissa oli myös muutos menossa, joten sen luotettavuus on kyseenalainen. Ennen viimeistä näytettä analysaattori käytiin kalibroimassa, jonka jälkeen IC4-pitoisuus saatiin lähelle laboratoriotulosta.

Vertailun perusteella analysaattorin poikkeama laboratoriotulokseen on tasainen, eikä elä yli 2 %-yksikköä.

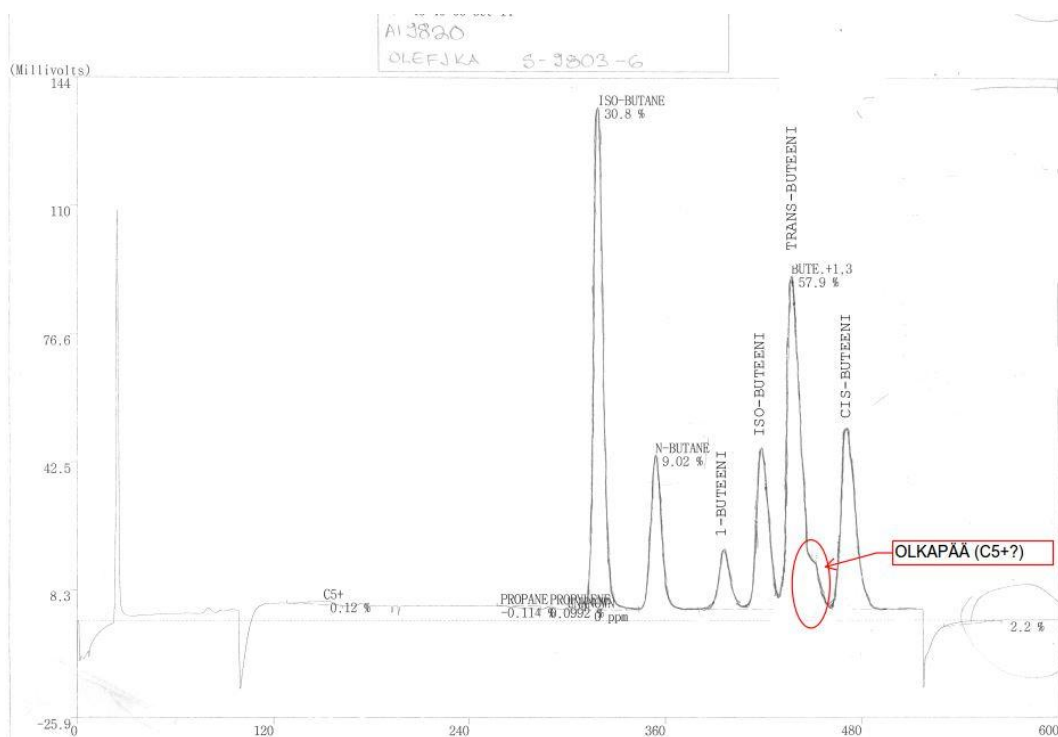
NC4-pitoisuuden pysyvä poikkeama voi johtua isobutaanikierron pienistä ASO-pitoisuuksista. ASO:n koostumus vaihtelee, mikä tekee sen mittaamisen vaikeaksi.

AI9820_1



Kuva 9. AI9820_1-analysaattorin luotettavuusvertailu. Poikkeama ilmoitettu mol%-yksikköä analysaattori-laboratoriotulos.

Olefiinisyötön analysaattoria kalibroidiin ennen viidettä näytettä vastaamaan laboratoriotuloksia. Kuvaajasta voi nähdä, että analysaattorin ja laboratoriotuloksen välinen ero ei elä yli 1 %-yksikön.



Kuva 10. AI9820:n grammi MTBE:n häiriön yhteydessä.

Analysaattori laskee buteenien, olefiinien, kokonaispitoisuuden. Vertailunäytteitä analysoitaessa laboratoriossa ihmeteltiin olefiinisytössä esiintyvää poikkeamaa. Kromatografian piirtämässä kuvaajassa esiintyi jokin tuntemattoman aineen aiheuttama ”olkapää” Trans-buteenien kyljessä, joka mahdollisesti vääristää sekä analysaattorin että laboratorion tulosta. Päädyimme siihen lopputulokseen, että tuo aine voisi olla buteenia raskaampaa hiilivetyjakeita, C5+, jota normaalisti saadaan alkyloinnin syöttöön vain vähäisiä pitoisuuksia, alle 1 mol-%. Olkapään vaikutus olefiinijakeiden kokonaismäärään on suuruudeltaan 2-3 mol-%. Analysaattori laski nämä osaksi isobuteeneja, jolloin I/O-suhdelaskenta antoi todellista huonomman tuloksen. (Pettinen 2015.)

MTBE-yksikkö poistaa normaalitilanteessa lähes kaiken isobuteenin ALKY:n syötöstä. Tämän vuoksi sitä ei myöskään mitata erikseen alkyloinnissa. MTBE:ssä oleva analysaattori AI1904 näytti, että osa isobuteeneista jäi reagoimatta ja pääsi ALKY:n syöttöön. Käytännössä tämä tarkoitti vaihtelevasti 5-12 mol-% isobuteenia olefiinisytössä sekä epämääräisiä pitoisuuksia raskaampia, C5+, jakeita. C5+-jakeen määrä pyritään minimoimaan alkyloinnin syötössä, koska sen tiedetään tuottavan ASO:a.

5.3 Suhdesäädön muutokset

Ennen koeajoa jokaiselta vuorolta kysyttiin, mitkä ovat heidän mielestään I/O-suhdesäädön kannalta merkittävät pullonkaulat ja onko heillä mielessä jotain, mitä haluaisivat testata koeajon yhteydessä.

Haastatteluissa esille tulleita asioita:

- Selkeyttimen FA-9807 pinta voi rajoittaa olefiinisyötön nostoa.
- Mikä on riittävä propeenipitoisuus IC4-kierrossa, jotta DA-9802:n paine pysyy vakaana?
- Saadaanko n-C4-pitoisuutta laskettua IC4-kierrossa?
- Muuttuuko alkylaatin koostumus, jos alasuuttimien I/O-suhdetta nostetaan?

Koeajo suunniteltiin ja toteutettiin yhdessä operoivien vuorojen sekä yksikön käyttöinsinöörin kanssa liitteessä 2 olevan suunnitelman mukaisesti. (Haastattelut Halijoki, Lah-tela, Riiski, Törmänen, Vahanen, Venäläinen.)

I/O-suhteen alaraja laskettiin käytönvalvojan luvalla 8:sta 7,5:een ennen koeajoja. Alarajan laskuun päädyttiin sen jälkeen, kun mittauksissa sekä TOP-laskennassa esiintyneet virheet oli saatu korjattua ja laskenta saatiin täsmäämään Excel-laskurin kanssa.

6 Päätelmät

6.1 Virtausmittausten toiminta

Perinteinen paine-eroon perustuva virtausmittaus on yksi toimintavarmimpia tapoja mitata virtausta öljynjalostamolla. Mitattavan aineen epäpuhtaudet ovat pääsääntöisesti tiedossa, eikä alkyloinnin syöttövirtauksissa yleensä ilmene impulssiputkia tukkivia partikkeleita.

I/O-suhdesäädössä mukana olevilta virtauksilta vaaditaan tavallista suurempaa tarkkuutta. FCA9809:n tapauksessa yksikön kokonaissyöttöä saatiin nostettua mittauksen puhallutuksen jälkeen I/O-suhteen noustua.

Virtausmittausten viritysparametrit tarkastettiin laskettamalla Neste Jacobsilla uudet viritysparametrit nykyisillä syöttöainepitoisuuksilla. Uudet parametrit poikkesivat vanhoista hyvin vähän, joten mahdollisen laskentavirheen aiheuttama poikkeama mittaus-tulokseen saatiin suljettua pois.

Kokonais-I/O-suhdelaskennassa mukana olevilla mittauksilla olisi syytä harkita säännöllisen huolto-ohjelman käyttöönottoa, jolloin virtausmittauksiin hitaasti tuleva poikkeama pysyisi mahdollisimman pienenä.

Syöttövirtausten, FCA9809 ja FCA9812, mittauksiin mahdollisesti syntyvä mittavirhe voidaan jatkossa tunnistaa myös I/O-Excelillä, jossa on erikseen laskettu tuoresyötön I/O-suhde. Tämän tulisi olla noin $1,1 \text{ m}^3/\text{m}^3$.

6.2 Analysaattoreiden toiminta

Analysaattoreiden skaalauksen käyttöönotto helpotti tuotannon työskentelyä, varsinkin kun vertailunäytteillä kyettiin osoittamaan analysaattoreiden tarkkuus.

Olefiinisyötön analysaattorin virhettä epäiltäessä on aihetta tarkistaa, kulkeutuuko MTBE- yksiköstä isobuteeneja normaalia enemmän. Yli 5 mol-%:n isobuteenipitoisuus olefiinisyötössä antaa viitteitä häiriöstä, jonka seurauksena yksikköön voi päätyä myös

haitallisia C5+-jakeita. Tämä puolestaan tukkii prosessilaitteita ja voi aiheuttaa nopeasti suuria vääristymiä IC4-kierron virtausmittauksiin ja AI9810_2:n pitoisuuksiin.

IC4-kierto on noin 6-kertainen verrattuna tuoreisobutaanin syöttöön, joten AI9810_2:lla on suuri merkitys koko I/O-suhdelaskennan kannalta. Jo yhden prosenttiyksikön muutoksella isobutaanikierron puhtaudessa on vaikutusta I/O-suhdelaskentaan. TOP-järjestelmässä olevaa laskennan oikeellisuutta tarkastellessa kävi ilmi, että AI9810_2:n tulos oli poistettu I/O-suhdelaskennasta ja korvattu laboratoriotuloksella. Laboratorioanalyysien tarkasteluväli vaihteli yhdestä vuorokaudesta useisiin vuorokausiin. Prosessi elää kuitenkin kellon ympäri, joten operaattorit saattoivat tietämättään ajaa prosessin huonoon tilaan näytteiden pitkästä analyysivälisestä johtuen.

AI9810_2 otettiin riskitarkastelun (PTM) jälkeen mukaan I/O-suhdelaskentaan ja koeajot voitiin suorittaa reaaliaikaisten analyysien perusteella. Koeajon aikana viedyt vertailunäytteet osoittivat, että analysaattorit näyttävät riittävän luotettavasti.

Analysaattoreiden normaalit pitoisuudet yksikön ollessa täydellä syötöllä on esitetty liitteessä 1.

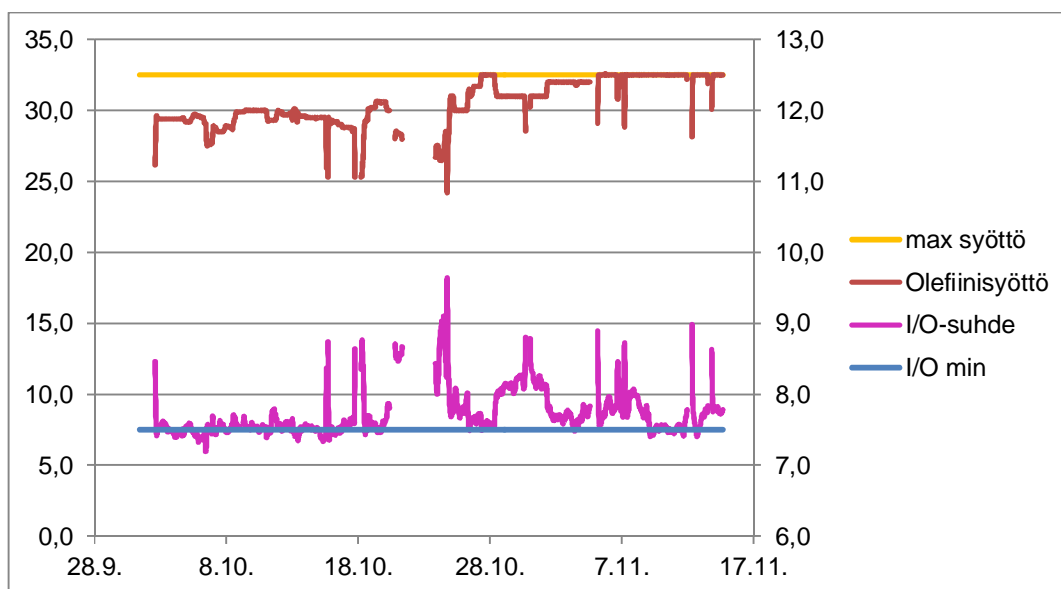
6.3 Koeajot

Varsinainen alkylointiprosessin koeajojakso kesti 5 vuorokautta, jonka aikana tehtiin yksi muutos jokaisen alkavan 12 tunnin työvuoron aikana. Jokaisen muutoksen jälkeen vietiin laboratorioon vertailunäytteet alkylaalista. Lisäksi IC4-kierrosta vietiin näytteet kierron pitoisuuksien muutoksen yhteydessä. Koeajo- ja aikataulusuunnitelma ovat liitteessä 3.

Koeajon alku sujui loistavasti, ensimmäiset muutokset sekä niille asetetut tavoitteet saavutettiin aikataulussa. IC4-kierron puhdistamisella tavoiteltiin korkeampaa I/O-suhdetta, joka myös toteutui. Tämä huomattiin myös tuotannon suunnittelussa, jossa haluttiin nostaa yksikön syöttö maksimitasolle heti kun mahdollista. Tämä tehtiin siitä huolimatta, että olimme sopineet ajavamme 5 vuorokauden koeajon tasaisella syötöllä. Syötön noston lisäksi koeajon lopun teki haastavaksi MTBE:n syöttöön otettu lisäsyöttö, raffinaatti, jonka tiedettiin runsaalla normaalibutaanipitoisuudellaan heikentävän I/O-suhdetta.

Taulukko 1. IC4-kierron puhdistuskoeajo

	t/h	Alkutilanne	Tavoite	Lisäsyötöllä Toteutunut	
Olefiinisyöttö	29,5			32,5 t/h	
IC-kierto	t/h 130	mol-%	mol-%		
Propaani		8,20 %	6,00 %	7,00 %	analysaattorin mukaan 6%
IC4		75,40 %	82,00 %	78,00 %	
NC4		13,40 %	10,00 %	12,00 %	
C5+		3,00 %	2,00 %	3,00 %	analysaattorin mukaan 4%
YHT		100,00 %	100,00 %	100,00 %	
butaanin ulosotto DA-9802:sta	t/h 3,1	mol-%	mol-%	4,5 t/h	
IC4		1,00 %	5,00 %	6,00 %	
NC4		82,80 %	85,00 %	83,00 %	
C5+		16,20 %	10,00 %	11,00 %	
YHT		100,00 %	100,00 %	100,00 %	
Alkylaattituotanto	33 t/h		?	36 t/h	



Kuva 11. ALKY:n olefiinisyötön määrän muutos koeajojen yhteydessä.

Trendien perusteella on helppo todeta, että ALKY saatiin täydelle 32,5 t/h syöttötasolle koeajon alkuvaiheessa toteutetun IC4-kierron säädön yhteydessä. Lisäksi kesken koeajon MTBE:n syöttöön otettu raffinaatti pystyttiin käsittelemään myös ALKY:ssa, IC4-kierron puhdistumisen myötä. Trendissä näkyvä marraskuun alun 2014 syötön lasku johtui MTBE-yksikön häiriöistä, jonka jälkeen ALKY:ssa päästiin jälleen maksimisyötölle.

Tulosten perusteella n-C4-ulosoton kasvattaminen vaikuttaa positiivisesti IC4-kierron pitoisuuksiin ja sitä kautta parantaa I/O-suhdetta.

Alasuuttimien koeajo suoritettiin 32 t/h syöttötasolla. Sen yhteydessä havaittiin säätöventtiilien avautuvan täysin auki, kun I/O-suhdetta alasuuttimille nostettiin yli 12. Pienemmillä syöttötasoilla alasuuttimien I/O-suhdetta on mahdollista nostaa. I/O-suhteen nostolla alasuuttimille 11 ylöspäin ei havaittu olevan vaikutusta alkylaatin laatuun.

6.4 Prosessihäiriöt

Työn alkuvaiheessa perehdytyksessä esiin nostettujen ongelmia selvittely osoittautui todella haastavaksi, koska ALKY:ssä tapahtui useita syötön laatupoikkeamista johtuvia prosessihäiriöitä.

ALKY kuuluu olefiiniketjuun, jossa olefiinisyöttö saadaan NKJT-yksiköstä.

Syöttöaines kulkee reittiä NKJT(TL2) -> MTBE(TL3) -> DBH(TL3) -> OKSY(TL1) -> ALKY(TL2).

Prosessihäiriötilanteissa eri tuotantolinjojen ja prosessiyksiköitä ohjaavien operaattoreiden välinen kommunikointi on erittäin tärkeää, jotta ketjussa viimeisenä olevassa yksikössä tapahtuvat häiriöt saadaan minimoitua.

MTBE-yksikön reaktoreiden heikko käsittelykapasiteetti käyntijakson lopulla mahdollisti isobuteenijakeen pitoisuuden kasvun alkyloinnin syötössä. Samassa yhteydessä syötön mukana tuli jotain raskaampaa hiilivetyjaetta, joka aiheutti poikkeaman olefiinisyötön analysaattoriin, minkä seurauksena I/O-suhdelaskenta näytti väärin. Isobuteeni ei ole haitallista alkyloinnin prosessin tai tuotteen laadun suhteen. Normaalitytilanteessakin MTBE:n ohi ajetaan pieni määrä olefiinisyöttöä, jolloin myös isobuteenia pääsee pieniä määriä ALKY:n syöttöön.

6.5 I/O-suhde-Excel

Osana insinööriytötä tehty Excel on toteutettu lisenssin haltijalta saatujen laskentakäytöjen mukaan. Laskuria käytettiin apuna analysointilaitteiden sekä virtausmittausten luotettavuuden selvityksen yhteydessä. Mittausten huollon jälkeen laskurin ja TOP-järjestelmän vertailu osoitti Excelin antavan kokonais-I/O-suhdelaskennasta saman tuloksen.

Ylä- ja alasuuttimien laskennassa on pieni ero TOP:n ja Excelin välillä. Tällä ei kuitenkaan liene suurta käytännön vaikutusta, koska ylä- ja alasuuttimien suhteita ei muuteta laskennan perusteella.

HUOM!! täytä vain laskurin PUNAISIIN (virtaukset) ja VIHREISIIN (pitoisuudet) kenttiin										
	t/h	mol-%	M kg/kmol	m	tiheys(15°C) kg/m³	V	V v-%	V m³/h	m kg/m³	m-%
Olefiinisyyttö	31,8				585,75			54,29		
Isobutaani		30,8	58,13	1790,40	562,98	3,18	32,72	17,76	184,18	31,44
Olefiinit		56	56,11	3142,16	600,00	5,24	53,87	29,25	323,24	55,18
n-Butaani		13,1	58,13	761,50	584,06	1,30	13,41	7,28	78,34	13,37
Butadieeni		0	54,09	0	627,30	0,00	0,00	0,00	0,00	0,00
Yht.		99,9				9,72	100,00	54,29	585,75	100,00
Tuoreisobutaani	9,30				559,29			16,63		
Isobutaani		88,4	58,13	5138,69	564,00	9,11	90,13	14,99	508,33	90,89
n-Butaani		1,2	58,13	69,76	585,00	0,12	1,18	0,20	6,90	1,23
Propani		10,1	44,10	445,41	507,00	0,88	8,69	1,45	44,06	7,88
Yht.		99,7				10,11	100,00	16,63	559,29	100,00
Kiertoisobutaani	126				566,74			222,32		
Isobutaani		77,6	58,13	4510,89	564,00	8,00	78,63	174,80	443,45	78,25
n-Butaani		11,9	56,11	667,71	585,00	1,14	11,22	24,95	65,64	11,58
Propani		6,1	44,10	269,01	507,00	0,53	5,22	11,60	26,45	4,67
raskaammat		4,4	72,15	317,46	632,00	0,50	4,94	10,98	31,21	5,51
Yht.		100,00				10,17	100,00	222,32	566,74	100,00
DA-9803:n IC4	6,4				558,09			11,47		
Isobutaani		81	58,13	4708,53	564,00	8,35	83,10	9,53	468,71	83,98
n-Butaani		5	56,11	280,55	585,00	0,48	4,77	0,55	27,93	5,00
Propani		14	44,10	617,40	507,00	1,22	12,12	1,39	61,46	11,01
Yht.		100				10,05	100,00	11,47	558,09	100,00
tuoresyötön jakautuminen		t/h								m³/h
yläsuuttimille	50,0 %	20,55								29,25
alasuuttimille	50,0 %	20,55								32,75
kiertosyötön jakautuminen		t/h								
yläsuuttimille	32,0 %	42,69								
alasuuttimille	68,0 %	90,71								
Kokonaissyötön I/O-suhde		7,42								m³/m³
Ylä ja keskisuuttimien I/O-suhde		5,15								m³/m³
kiertoisobutaani		58,99								Yläsuuttimien paikallinen I/O-suhde 25,57
tuoreisobutaani		16,37								keskisuuttimien paikallinen I/O-suhde 22,54
olefiinit		14,62								
Alasuuttimien I/O-suhde		9,69								m³/m³
kiertoisobutaani		125,35								
tuoreisobutaani		16,37								
olefiinit		14,62								
Tuoresyötön I/O-suhde		1,12								m³/m³

Kuva 12. I/O-laskentataulukko

7 Tulosten yhteenveto

Työn alkuvaiheessa tavoitteeksi asetettu I/O-suhdelaskennan avaaminen sekä mittaus-
ten luotettavuuden selvittäminen onnistui mielestäni hyvin. Tuotannon, kunnossapidon
sekä jalostamon laboratorion kanssa yhteistyössä löydettiin konkreettisia ratkaisuja
alkylointia vaivanneisiin ongelmiin. Yhtenä tärkeimmistä tuloksista voitaneen todeta
yksikön syöttötason nousu tasolta 30 t/h maksimitasolle 32,5 t/h.

Tuotannon suunnittelusta pyydetyn arvion mukaan alkylaatin tuotannon kasvulla saa-
vutettiin noin 300 k€ / kk voittoa. Syötön noston myötä välttyttiin MTBE:n lisäsyötön
myynniltä tappiolla ulos, josta olisi voinut seurata vastaavan suuruinen tappio (300 k€).

Virtausmittausten toiminta on pääosin hyvällä tasolla. Toki säännöllisellä huollolla,
esimerkiksi kuuden kuukauden välein, välttyttäisiin joiltakin prosessiongelmilta.

Skaalauksen käyttöönotolla analysaattoreiden tuloksista saatiin vertailukelpoisia labo-
ratoriotuloksien kanssa.

Olefiinisyötön analysaattorille olisi hyvä tehdä selvitys isobuteenien analysoinnin osalta
ja selvittää, voiko nykyisen analysaattorin konfiguroida näyttämään isobuteenipitoisuu-
den erikseen.

Liitteenä olevan I/O-suhdelaskurin koodin tutkiminen osoittautui todella haastavaksi jo
pelkästään sen pituuden vuoksi. Olisi hyvä selvittää laskurin muuttamista johonkin ny-
kyaikaisempaan ja helpommin ymmärrettävään versioon, jossa varsinainen laskenta on
tiivistettynä selkeästi erilleen muusta ohjelmoinnista.

Lähteet

Alkylointiyksikön yleisohje v.10. Neste Oyj.

Halijoki, Pertti. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 6.10.2014.

Lahtela, Jouko. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 12.10.2014.

Pettinen, Keijo. 2015. Laiteasiantuntija, Neste Oyj, Porvoo. Haastattelu. 10.11.2015.

Riiski, Vesa. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 5.10.2014.

Torkkel, Tuire. 2015. Kunnossapitoinsinööri, Neste Oyj, Porvoo. Haastattelu. 10.11.2015.

Törmänen, Aimo. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 8.10.2014.

Vahanen, esa. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 12.10.2014.

Venäläinen, Tero. 2014. Operaattori, Neste Oyj, Porvoo. Haastattelu. 8.10.2014.

Phillips Petroleum, Split Olefin Feed Technology(SOFT), Calculations

Sivonen, Markku. 1999. Teollisuuden instrumentointi: rakenne ja suunnittelu. Helsinki. AEL.

Analysaattoreiden pitoisuudet

Analysaattoreiden ilmoittamat pitoisuudet ideaalitalanteessa yksikön ollessa täydellä syötöllä 32,5 t/h.

AI9810_1	mol-%
Isobutaani	92
n-Butaani	1
Propaani	7
Yht.	100

AI9810_2	
Isobutaani	80
n-Butaani	10
Propaani	6
raskaammat	4
Yht.	100

AI9820_1	
Propaani	2
Isobutaani	31
Olefiinit	54
n-Butaani	13
Butadieeni	0
Yht.	100

Koeajosuunnitelma

NESTE OILTilapäisohje
Sisäinen

1 (1)

Elmeri Waljus

14.10.2014

ALKY:N IO-SUHTEEN SÄÄDÖN KOEAJO

1 Voimassaoloaika

27.10. - 30.11.2014

2 Kohderyhmä

Alkylöinnin tuotantohenkilökunta

3 Johdanto

Pertti Kärki on tekemässä lopputyötä IO-suhteen laskennasta. Tähän työhön liittyen suoritetaan koeajot, joilla pyritään saamaan IO-suhteen laskenta TOP:iin mahdollisimman todennukaiseksi. Samalla pyritään hakemaan IO-suhdesäätöön optimaaliset ajoarvot. Tämä voi mahdollistaa suuremman syöttötason yksikköön sekä vähentää ASO:n muodostumista selkeyttimessä.

4 Ohjeistus koeajoa varten

Koeajo aloitetaan, kun KTO5 on takaisin tuotannossa ja alkylöinti stabiilissa tilassa. Varmistetaan Kärjeltä aloitusaikataulu.

Koeajon aikataulu käy ilmi liitteenä olevasta excelistä.

Koeajossa pyritään nostamaan IC4-kierron IC4-pitoisuutta vähentämällä NC4:n ja C3:n määrää kierrossa.

Ensimmäisessä vaiheessa propaania vähennetään lisäämällä propaanin ulosottoa varastoon DA-9804:sta. Propaania otetaan 200kg/h enemmän kuin normaalitilanteessa, kunnes saavutetaan haluttu C3-pitoisuus IC4-kierrossa. Tavoitellaan 6 mol-% pitoisuutta C3:lle.

Toisessa vaiheessa NC4:n määrää vähennetään lisäämällä NC4-ulosottoa DA-9802:sta. Nostetaan NC4:n ulosottoa 500kg kerrallaan, kunnes NC4:n ulosoton IC4-pitoisuus on 5 mol-%. IC4-kierron NC4-pitoisuuden tavoite on 10 mol-%.

Näiden muutoksien avulla pyritään nostamaan IC4-kierron IC4-pitoisuus yli 80 mol-%:in.

Seuraavaksi raskaiden osuus NC4-ulosotossa pyritään laskemaan alle 10 mol-%. Joudataan mahdollisesti laskemaan uunin lämpöä 0,5 C kerrallaan.

Kun IC4-kierto on saatu puhdistettua, aloitetaan alasuuttimien IO-suhteen nosto. Tavoite alasuuttimien IO-suhteelle on 13. Tämä pyritään saavuttamaan laskemalla hitaasti alasuuttimille ajettavan olefiinisyötön määrää. Seurataan alkylaatin laatua sekä ASO:n muodostumista.

Koeajon aikana viedään ylimääräisiä näytteitä. Näiden näytteiden näytepyynnöt hoitaa Kärki.

Pertti Kärki 0452350804

Elmeri Waljus 0504582566

Koeajon aikataulu

Aikatauluus	27.10.2014	28.10.2014	29.10.2014	30.10.2014	31.10.2014	1.11.2014	2.11.2014
Proopaanin vähennys kierrosta	Klo 12-24	12-24	12-24	12-24	12-24	12-24	12-24
I _{C4} -pitoisuuden nosto butaanin ulosotossa							
Raskaiden vähentäminen butaanin ulosotossa							
Alasuuttimien I/O-suhteen nosto	10->11						
	11->12						
	12->13						

Olefiinisyytön laboratorioanalyysi 8.10.2014

IP473 NESTEKAASUJEN KOMPONENTIT

NÄYTE: 03476800 OLEFJKA

ANALYSOIJJA: System (cds_jabra)

RESULT SET: \\OPENLAB\CD\CD\CD_Data\Projects\GC10211\Result\Lokakuu 2014\03476800 OLEFJKA 2014-10-08 09-57-36 (GMT +03-00).rst\03476800 OLEFJKA 2014-10-08 09-57-36 (GMT +03-00).rst

Back Signal Results Komponentti	wt-%	Mol%	Area	Retentioaika
	0,0000		250	0,01
METAANI	0,0000 BDL			
ETAANI	0,0000 BDL			
ETEENI	0,0000 BDL			
PROPAANI	0,0154	0,019932	8111	5,89
SYKLOPROPAANI	0,0000 BDL			
PROPEENI	0,0000 BDL			
ISO-BUTAANI	32,2154	31,566902	17037905	9,70
N-BUTAANI	10,1749	9,970106	5410596	10,23
UNKNOWN	0,0210	0,020607	11183	11,66
TRANS-2-BUTEENI	25,9620	26,353018	14140836	12,31
1-BUTEENI	4,2912	4,355793	2363421	12,49
ISO-BUTEENI	12,4976	12,685837	6789722	12,78
CIS-2-BUTEENI	14,7604	14,982669	8150940	13,03
NEO-PENTAANI	0,0241	0,019042	12635	13,28
ISO-PENTAANI_P	0,0082	0,006453	4282	13,83
MAC	0,0000 BDL			
N-PENTAANI	0,0000 BDL			
1,3-BUTADIEENI	0,0000 BDL			
C6+RASK	0,0114	0,007525	5958	14,94
C6+RASK	0,0183	0,012115	9592	15,36
C6	0,0000 BDL			

Totals				
	100,0000	100,000000	53945431	

C6+RASK	0,0297	0,019641	15550
ISO-PENTAANI	0,0082	0,006453	4282
UNKNOWN	0,0210	0,020607	11183

Totals				
	0,0589	0,046701	31015	

C2+KEV	0,0000		0
C2YHT	0,0000		0
C3+KEV	0,0154		8111
C3YHT	0,0154		8111
C4YHT	99,9015		53893420
C4OLEF	57,5112		31444919
OLEFIINIT	57,5112		31444919
C4=+13C4=	57,5112		31444919

Totals				
	272,4660		148244399	

TOP -järjestelmän I/O-laskurin ohjelmakoodi

```

                                ALKYTCALC.txt
PROGRAM ALKYTCALC
C*****
C**
C**   This document contains proprietary information of SETPOINT,
C**   Inc., and is tendered subject to the condition that no copy
C**   or other reproduction be made in whole or in part for use
C**   other than client's own internal use, and that no use be made
C**   of information herein except for the purpose for which it is
C**   transmitted, without express written permission of SETPOINT,
C**   Inc.
C**
C*****
C+
C*****
C
C> Module Name ----- ALKYTCALC.PRF
C> Title ----- Alkylation Process Unit (18), Process Area 2
C> Source Language - FORTRAN 90
C
C> Abstract:
C-----
C
C   This is the General Area Calculation part for the Alkylation unit.
C
C> Revision History:
C-----
C
C   DATE           BY           REASON
C-----
C
C   28-FEB-2000 Björn Hamström   New calculation program
C   27-JUN-2000 Marja Sarikkola   Added PCT calculation for ALKYT_D01PCT07
C   28-JUN-2000 Björn Hamström   Added logic around type 35 constraint
C                                   controllers
C   06-JUL-2000 Marja Sarikkola   Added calculation of ALKYT_D01PCT32
C   10-JUL-2000 Marja Sarikkola   Added feed and product composition
C                                   calculations, mass balance calculations,
C                                   I/O-ratio calculations and reflux/dist-
C                                   ratio calculations
C   20-JUL-2000 Marja Sarikkola   Changed ALKYT_D01PCT07 to ALKYT_D01PCT09,
C                                   PCT calculated with empirica linear
C                                   correlation instead of PFPCT-routine
C   20-JUL-2000 Marja Sarikkola   Added calculation for additional IC4 feed
C                                   required (ALKYT_LISAIC4)
C   27-JUL-2000 Marja Sarikkola   Added calculation of ALKYT_PDICVE
C   15-AUG-2000 Björn Hamström   Added 3 ALKYCMI watchdogs
C   07-SEP-2000 Björn Hamström   Removed pc type 35 calculations and
C                                   changed the G2 watchdog logic
C   26-SEP-2000 Marja Sarikkola   Removed ALKYT_D01PCT09 and ALKYT_D01PCT32
C                                   calculations
C   14-NOV-2000 Björn Hamström   Added ALKYCMI_F9885_WDOG watchdog
C   30-JAN-2001 Björn Hamström   Removed unused sections like IRV logic etc.
C   01-FEB-2001 Björn Hamström   Cleanup
C   30-AUG-2004 Björn Hamström   Removed ALKYCMI_F9885_WDOG watchdog
C   01-OCT-2004 Björn Hamström   Corrected VDO data type to INTEGER*4
C   21-DEC-2005 Björn Hamström   Changed AI_9810_2_C3 -> ALKY_IC42_C3
C                                   AI_9810_2_IC4 -> ALKY_IC42_IC4
C   17-SEP-2008 Björn Hamström   Changed all Alcont tagnames to new DNA names
C   16-OCT-2008 Jaakko Lämsä     Changed PCT calculation. PCA9802 engineering
C                                   unit changed bar > kPa
C-----

```

```

C
C                                     ALKYTCALC.txt
C
C> Externals          Description
C-----
C P DBGETF           Floating Variable Access
C P DBGETFF          Filter Factor Variable Access
C P DBGETI           Integer Variable Access
C P DBSETF           Floating Variable Write
C P DBSETI           Integer Variable Write
C P DBTEXT           Get tagname, description and engineering unit
C P MESSAGES         Alarm Message Routine
C P OISETST          Set controller status
C T FILTRS           Variable Filtering
C T DIVCHK           Divide by Zero Check and Divide
C T PLABEL           Place a Label
C T PFPCT            Pressure Compensated Temperature
C T BALYD            Product Yields and Material Balance Closure
C S SYSSHIBER        Hibernate the Calling Process
C [P=PMS, S=System, T=Techlib]
C
C
C> Messages used in this program:
C-----
C
C 730 ALKYTCALC VIRHENUMERO XXX TIETOKANTAKUTSUSTA IFERR=XXX
C
C
C See the file PMS520:<SETPOINT.DATA>SP_MESSAGES.MES for a complete
C list of all messages defined.
C-----
C
C IMPLICIT NONE
C-----
C EXTERNAL SUBROUTINES/FUNCTIONS
C-----
C
C REAL*4    FILTRS
C REAL*4    DIVCHK
C-----
C LOCAL VARIABLES
C-----
C
C - declarations for local indices, flags, variables, etc...
C INCLUDE 'VS_LAYER_INCLUDE:VSCOMMON.INC'
C ... ( Declares JPRINT, ILUNUM, IDETAL (debug level), and IERROR )
C
C - declare the Group, Area and Alarm variables
C
C INTEGER*4  IDBPG      ! PMS Database Group
C INTEGER*4  IDBPA      ! PMS Database Area
C INTEGER*4  IURGALARM ! PMS Database Message Type
C PARAMETER (IDBPG = 18) ! group = 18, ALKY unit
C PARAMETER (IDBPA = 2)  ! area = 2, Production line 2
C PARAMETER (IURGALARM = 3) ! Message Type, 3=Urgent
C
C - declare variables
C
C INTEGER*4  I          ! iteration index
C INTEGER*4  IWK         ! iteration index
C INTEGER*4  JDETAL     ! level flag minus 1
C INTEGER*4  ISTATUS    ! internal tagname status variable
C INTEGER*4  JPRNT      ! internal print flag
C
C Page 2

```

```

                                ALKYTCALC.txt
INTEGER*4  IFERR                ! PMS DB error flag
INTEGER*4  MERR                 ! PMS DB error flag
C
INTEGER*4  IERROR1              ! Intermediate Error Flag Holder
INTEGER*4  IERROR2              ! Intermediate Error Flag Holder
INTEGER*4  IERR                 ! Intermediate Error Flag Holder
C
INTEGER*4  VALUE                !Intermediate value digitals
C
INTEGER*4  GOOD_STATUS,        BAD_STATUS
INTEGER*4  GOOD_IER,           BAD_IER
PARAMETER ( GOOD_STATUS = 1,  BAD_STATUS = 0 )
PARAMETER ( GOOD_IER   = 0,  BAD_IER   = 1 )
C
INTEGER*4  TRUE_VALUE,         FALSE_VALUE
PARAMETER ( TRUE_VALUE  = 1,  FALSE_VALUE = 0 )
C
INTEGER*4  FILTYP              ! Filtering Module Factor Calc Flag
INTEGER*4  INIEXI              ! Filtering Module Initialization Flag
C
REAL*4     NUMER                ! DIVCHK Numerator
REAL*4     DENOM                ! DIVCHK Denominator
C
REAL*4     FLOW(10)
REAL*4     Y(10)
C
INTEGER*4  SID(10)
-----
C
C
C  - declare the molecular weights (kg/kmol)
C
REAL*4     MWC3H8                ! PROPANE
REAL*4     MWC3H6                ! PROPYLENE
REAL*4     MWNC4                 ! n-BUTANE
REAL*4     MWIC4                 ! ISOBUTANE
REAL*4     MWC4T                 ! BUTENE
C
PARAMETER ( MWC3H8 = 44.0962 )
PARAMETER ( MWC3H6 = 42.0804 )
PARAMETER ( MWNC4  = 58.1230 )
PARAMETER ( MWIC4  = 58.1230 )
PARAMETER ( MWC4T  = 56.1072 )
C
-----
C
C
C  - declare the densities (kg/m3)
C
REAL*4     DNC3H8                ! PROPANE
REAL*4     DNC3H6                ! PROPYLENE
REAL*4     DNNC4                 ! n-BUTANE
REAL*4     DNIC4                 ! ISOBUTANE
REAL*4     DNC4T                 ! BUTENE
C
PARAMETER ( DNC3H8 = 507.7 )
PARAMETER ( DNC3H6 = 522.0 )
PARAMETER ( DNNC4  = 584.4 )
PARAMETER ( DNIC4  = 563.1 )
PARAMETER ( DNC4T  = 610.0 )
C
-----
C
C
C  - the variables needed for the calculations
C
REAL*4     VSUM                  ! total volume of olefin feed
REAL*4     VC4T                  ! volume fraction of C4T in olefin feed

```

```

                                ALKYTCALC.txt
REAL*4 VIC4           ! volume fraction of IC4 in olefin feed
REAL*4 VNC4           ! volume fraction of NC4 in olefin feed
REAL*4 VC3H6          ! volume fraction of C3H6 in olefin feed
REAL*4 VC3H8          ! volume fraction of C3H8 in olefin feed
REAL*4 MSUM           ! total mass of olefin feed
REAL*4 MC4T           ! mass fraction of C4T in olefin feed
REAL*4 MIC4           ! mass fraction of IC4 in olefin feed
REAL*4 MNC4           ! mass fraction of NC4 in olefin feed
REAL*4 MC3H6          ! mass fraction of C3H6 in olefin feed
REAL*4 MC3H8          ! mass fraction of C3H8 in olefin feed
REAL*4 DEN            ! density of olefin feed
C
REAL*4 VSUM9809       ! total volume of isobutane feed
REAL*4 V9809IC4       ! volume fraction of IC4 in isobutane feed
REAL*4 V9809NC4       ! volume fraction of NC4 in isobutane feed
REAL*4 V9809C3H8      ! volume fraction of C3H8 in isobutane feed
REAL*4 MSUM9809       ! total mass of isobutane feed
REAL*4 M9809IC4       ! mass fraction of IC4 in isobutane feed
REAL*4 M9809NC4       ! mass fraction of NC4 in isobutane feed
REAL*4 M9809C3H8      ! mass fraction of C3H8 in isobutane feed
REAL*4 DEN9809        ! density of isobutane feed
C
REAL*4 VOLEF          ! total olefin (C3+C4) volume flow in olefin feed
REAL*4 VOLEFA         ! olefin volume flow to GD-9801A
REAL*4 VOLEFB         ! olefin volume flow to GD-9801B
REAL*4 MFOLEF         ! olefin mole flow
C
REAL*4 MW9809         ! MW of isobutane feed
REAL*4 MW9812         ! MW of olefin feed
REAL*4 MW9820         ! MW of recycled isobutane
REAL*4 MPIC43         ! mole fraction of IC4 in recycled IC4 from DA-98
REAL*4 MW9843         ! MW of recycled IC4 from DA-9803
C
REAL*4 MFIC4          ! mole flow of IC4 in isobutane feed
REAL*4 MPKIC4         ! mole flow of IC4 in olefin feed
REAL*4 MPKIC4         ! mole flow of IC4 in recycled IC4
REAL*4 MPKIC4         ! mole flow of IC4 in recycled IC4 from DA-9803
REAL*4 MFIC4A         ! fresh IC4 mole flow to GD-9801A
REAL*4 MPKIC4A        ! recycled IC4 mole flow to GD-9801A
REAL*4 MFIC4B         ! fresh IC4 mole flow to GD-9801B
REAL*4 MPKIC4B        ! recycled IC4 mole flow to GD-9801B
C
REAL*4 VAIC4          ! total IC4 volume flow to reactor
REAL*4 VAIC4A         ! IC4 volume flow to GD-9801A
REAL*4 VAIC4B         ! IC4 volume flow to GD-9801B
REAL*4 IC4             !
REAL*4 K               !
REAL*4 RATIO          !
C
C - the statuses needed for the calculations
C
INTEGER*4 MC4T_ST
INTEGER*4 MIC4_ST
INTEGER*4 MNC4_ST
INTEGER*4 MC3H6_ST
INTEGER*4 MC3H8_ST
INTEGER*4 VC4T_ST
C
INTEGER*4 M9809IC4_ST
INTEGER*4 M9809NC4_ST
INTEGER*4 M9809C3H8_ST
C
INTEGER*4 DEN_ST
INTEGER*4 DEN9809_ST
C
INTEGER*4 VOLEF_ST
INTEGER*4 VOLEFA_ST
INTEGER*4 VOLEFB_ST
C

```

```

ALKYTCALC.txt
INTEGER*4 MW9809_ST
INTEGER*4 MW9812_ST
INTEGER*4 MW9820_ST
C
INTEGER*4 MFIC4_ST
INTEGER*4 MFKIC4_ST
INTEGER*4 MFKKIC4_ST
INTEGER*4 MFKKKIC4_ST
INTEGER*4 MFIC4A_ST
INTEGER*4 MFKKIC4A_ST
INTEGER*4 MFIC4B_ST
INTEGER*4 MFKKIC4B_ST
C
INTEGER*4 VAIC4_ST
INTEGER*4 VAIC4A_ST
INTEGER*4 VAIC4B_ST
C
INTEGER*4 IC4_ST
INTEGER*4 K_ST
C
-----
REAL*4 F9843_IC4M ! variable used as a substitute
PARAMETER ( F9843_IC4M = 85.0 ) ! for lab result ALKY_IC43_IC4M
C
=====
C
C - VMS / SYSTEM VARIABLES
C
-----
C
C - variables used to put this process to sleep
C
INTEGER*4 ICOND ! error return from SYS$HYBER
INTEGER*4 SYS$HYBER ! hibernates the process
C
C - variables for timestamp
C
CHARACTER*8 CDATE
CHARACTER*10 CTIME
C
=====
C
C - Number of Inputs by Type
C
INTEGER*4 IAIMX ! instrument inputs (AI's)
INTEGER*4 ILVMX ! instrument limit values (LV's)
INTEGER*4 IFMEMX ! floating point manual entries (FME's)
INTEGER*4 ICAMX ! calculated values (CA's)
INTEGER*4 IIMEMX ! integer manual entries (IME's)
INTEGER*4 IFVMX ! filter calculated variables (FV's)
INTEGER*4 IDIMX ! digital input variables (DI'S)
INTEGER*4 IDOMX ! digital output variables (VDO'S)
INTEGER*4 ICTMX ! controller targets (PC's)
C
-----
C
PARAMETER (IAIMX = 35)
PARAMETER (ILVMX = 0)
PARAMETER (IFMEMX = 0)
PARAMETER (ICAMX = 50)
PARAMETER (IIMEMX = 3)
PARAMETER (IFVMX = 0)
PARAMETER (IDIMX = 0)
PARAMETER (IDOMX = 0)
PARAMETER (ICTMX = 0)

```

```

C
C
C-----
C - tagname descriptions
C
CHARACTER*60 AITEXT(IAIMX) ! instrument inputs (AI's)
CHARACTER*60 LVMITEXT(ILVMX) ! instrument limit values (LV's)
CHARACTER*60 LVMAITEXT(LVMA) ! instrument limit values (LV's)
CHARACTER*60 LVCSTEXT(LVCS) ! instrument limit values (LV's)
CHARACTER*60 FMETEXT(IPMEMX) ! floating point manual entries (FME's)
CHARACTER*60 CATEXT(ICAMX) ! calculated values (CA's)
CHARACTER*60 IMETEXT(IIMEMX) ! integer manual entries (IME's)
CHARACTER*60 FVTEXT(IFVMX) ! filter calculated variables (FV's)
CHARACTER*60 DITEXT(IDIMX) ! digital input variables (DI's)
CHARACTER*60 DOTEXT(IDOMX) ! digital output variables (VDO's)
CHARACTER*60 CTTEXT(ICTMX) ! controller targets (PC's)
C
C-----
C - database values
C
REAL*4 AI(IAIMX) ! measurements
REAL*4 LVMI(ILVMX) ! max limit
REAL*4 LVMA(ILVMA) ! min limit
REAL*4 LVCS(ILVCS) ! max change limit
REAL*4 FME(IPMEMX) ! manual entry
REAL*4 CA(ICAMX) ! calculated value
INTEGER*4 IME(IIMEMX) ! integer value
REAL*4 FF(IFVMX) ! filter factors
INTEGER*4 DBFE(IFVMX) ! filtering enable bit - must be disabled
REAL*4 FV(IFVMX) ! filtered value
INTEGER*4 VDO(IDOMX) ! digital output value.
C Note DO reserved for Fortran !
INTEGER*4 INT_VDO(IDOMX) ! digital output value to interface routine
INTEGER*4 DI(IDIMX) ! digital input value
REAL*4 CT(ICTMX) ! controller target
C
C-----
C - database indices
C
INTEGER*4 IAI(IAIMX) ! instrument inputs (AI's)
INTEGER*4 ILV(ILVMX) ! Instrument limit values (LV's)
INTEGER*4 IPME(IPMEMX) ! floating point manual entries (FME's)
INTEGER*4 ICA(ICAMX) ! calculated values (CA's)
INTEGER*4 IIME(IIMEMX) ! integer manual entries (IME's)
INTEGER*4 IFV(IFVMX) ! filter calculated variables (FV's)
INTEGER*4 IDI(IDIMX) ! Digital input variables (DI's)
INTEGER*4 IDO(IDOMX) ! Digital output variables (VDO's)
INTEGER*4 ICT(ICTMX) ! controller targets (PC's)
C
C-----
C - tagname statuses
C
INTEGER*4 IAIST(IAIMX) ! instrument inputs (AI's)
INTEGER*4 IPMEST(IPMEMX) ! floating manual entries (FME's)
INTEGER*4 ICAST(ICAMX) ! calculated values (CA's)
INTEGER*4 IDOST(IDOMX) ! digital outputs (DO's)
INTEGER*4 IDIST(IDIMX) ! digital inputs (DI's)
INTEGER*4 IFVST(IFVMX) ! filter calculated variables (FV's)
C
C-----
C - variables used to access controller records
C
INTEGER*4 ICT_DBCSS(ICTMX) ! control loop state
INTEGER*4 ICT_PCTP(ICTMX) ! control loop test output
C
REAL*4 ICT_DBCV(ICTMX) ! control loop target value
INTEGER*4 ICT_ISTA(ICTMX) ! control loop status
REAL*4 RCT_DBVMAX(ICTMX) ! target value min

```



```
ALKYTCALC.txt
INTEGER*4 AI_9810_1_NC4_ST
INTEGER*4 ALKY_IC42_C3_ST
INTEGER*4 ALKY_IC42_IC4_ST
INTEGER*4 AI_9810_3_IC4_ST
INTEGER*4 TI9808_ST
INTEGER*4 PCA9802_ST
INTEGER*4 TDI9806_ST
INTEGER*4 FC9801_ST
INTEGER*4 FC9805_ST
INTEGER*4 FC9805_VE_ST
INTEGER*4 FC9806_ST
INTEGER*4 FC9836_ST
INTEGER*4 FC9845_ST
INTEGER*4 FC9849_ST
INTEGER*4 FC9852_ST
INTEGER*4 FCA9804_ST
INTEGER*4 FCSA9837_ST
INTEGER*4 FCSA9838_ST
INTEGER*4 TI9865_ST
INTEGER*4 FCA9812_VE_ST
INTEGER*4 FCA9809_VE_ST
INTEGER*4 PDC9804_VE_ST

- instrument input max limit

- instrument input min limit

- instrument rate of change limit

- calculated variable values
REAL*4 ALKYT_IC40LE
REAL*4 ALKYT_IC40LEA
REAL*4 ALKYT_IC40LEB
REAL*4 ALKYT_SYOTIC4
REAL*4 ALKYT_SYOTC4
REAL*4 ALKYT_SYOTC3
REAL*4 ALKYT_SYOTYHT
REAL*4 ALKYT_SYOTPRA
REAL*4 ALKYT_SYOTNC4
REAL*4 ALKYT_SYOTC30S
REAL*4 ALKYT_SYOTPRA0S
REAL*4 ALKYT_SYOTC40S
REAL*4 ALKYT_SYOTIC40S
REAL*4 ALKYT_SYOTNC40S
REAL*4 ALKYT_OSYHT
REAL*4 ALKYT_HC11
REAL*4 ALKYT_HC12
REAL*4 ALKYG_WDOG
REAL*4 ALKYT_D01PCT09
REAL*4 ALKYT_IC4VAR
REAL*4 ALKYT_TU0TC4
REAL*4 ALKYT_NC4VAR
REAL*4 ALKYT_C3VAR
REAL*4 ALKYT_TU0TALKY
REAL*4 ALKYT_F98060S
REAL*4 ALKYT_IC4VAR0S
REAL*4 ALKYT_F98090S
REAL*4 ALKYT_C3VAR0S
REAL*4 ALKYT_NC4VAR0S
REAL*4 ALKYT_F98360S
REAL*4 ALKYT_TU0TC40S
REAL*4 ALKYT_TU0TALKY0S
REAL*4 ALKYT_D01RF2DIST
REAL*4 ALKYT_D02RF2DIST
REAL*4 ALKYT_LISAIC4
```

```
ALKYTCALC.txt
REAL*4   ALKYT_IC4F_MY
REAL*4   ALKYT_OLEFF_MY
REAL*4   ALKYT_NC4_MY
REAL*4   ALKYT_ALKY_MY
REAL*4   ALKYT_PRA_MY
REAL*4   ALKYT_D01F_MY
REAL*4   ALKYT_D01_NC4_MY
REAL*4   ALKYT_D01_IC4VAR_MY
REAL*4   ALKYT_D01_IC4_MY
REAL*4   ALKYT_MB_CL
REAL*4   ALKYT_D01MB_CL
REAL*4   ALKYT_PDICVE
REAL*4   ALKYCMI_F9802_WDOG
REAL*4   ALKYCMI_F9804_WDOG
REAL*4   ALKYCMI_F9803_WDOG

C
C - calculated variable statuses
C
INTEGER*4   ALKYT_IC40LE_ST
INTEGER*4   ALKYT_IC40LEA_ST
INTEGER*4   ALKYT_IC40LEB_ST
INTEGER*4   ALKYT_SYOTIC4_ST
INTEGER*4   ALKYT_SYOTC4_ST
INTEGER*4   ALKYT_SYOTC3_ST
INTEGER*4   ALKYT_SYOTVHT_ST
INTEGER*4   ALKYT_SYOTPRA_ST
INTEGER*4   ALKYT_SYOTNC4_ST
INTEGER*4   ALKYT_SYOTC305_ST
INTEGER*4   ALKYT_SYOTPRA05_ST
INTEGER*4   ALKYT_SYOTC405_ST
INTEGER*4   ALKYT_SYOTIC405_ST
INTEGER*4   ALKYT_SYOTNC405_ST
INTEGER*4   ALKYT_0SYHT_ST
INTEGER*4   ALKYT_HC11_ST
INTEGER*4   ALKYT_HC12_ST
INTEGER*4   ALKYG_WDOG_ST
INTEGER*4   ALKYT_D01PCT09_ST
INTEGER*4   ALKYT_IC4VAR_ST
INTEGER*4   ALKYT_TUOTC4_ST
INTEGER*4   ALKYT_NC4VAR_ST
INTEGER*4   ALKYT_C3VAR_ST
INTEGER*4   ALKYT_TUOTALKY_ST
INTEGER*4   ALKYT_F980605_ST
INTEGER*4   ALKYT_IC4VAR05_ST
INTEGER*4   ALKYT_F980905_ST
INTEGER*4   ALKYT_C3VAR05_ST
INTEGER*4   ALKYT_NC4VAR05_ST
INTEGER*4   ALKYT_F983605_ST
INTEGER*4   ALKYT_TUOTC405_ST
INTEGER*4   ALKYT_TUOTALKY05_ST
INTEGER*4   ALKYT_D01RF2DIST_ST
INTEGER*4   ALKYT_D02RF2DIST_ST
INTEGER*4   ALKYT_LISAIC4_ST
INTEGER*4   ALKYT_IC4F_MY_ST
INTEGER*4   ALKYT_OLEFF_MY_ST
INTEGER*4   ALKYT_NC4_MY_ST
INTEGER*4   ALKYT_ALKY_MY_ST
INTEGER*4   ALKYT_PRA_MY_ST
INTEGER*4   ALKYT_D01F_MY_ST
INTEGER*4   ALKYT_D01_NC4_MY_ST
INTEGER*4   ALKYT_D01_IC4VAR_MY_ST
INTEGER*4   ALKYT_D01_IC4_MY_ST
INTEGER*4   ALKYT_MB_CL_ST
INTEGER*4   ALKYT_D01MB_CL_ST
INTEGER*4   ALKYT_PDICVE_ST
INTEGER*4   ALKYCMI_F9802_WDOG_ST
INTEGER*4   ALKYCMI_F9804_WDOG_ST
INTEGER*4   ALKYCMI_F9803_WDOG_ST

C
```



```

                                ALKYTCALC.txt
EQUIVALENCE ( AI( 29), FCA9804 )
EQUIVALENCE ( AI( 30), FCSA9837 )
EQUIVALENCE ( AI( 31), FCSA9838 )
EQUIVALENCE ( AI( 32), TI9865 )
EQUIVALENCE ( AI( 33), FCA9812_VE )
EQUIVALENCE ( AI( 34), FCA9809_VE )
EQUIVALENCE ( AI( 35), PDC9804_VE )
C
C
C - equivalence instrument input statuses
EQUIVALENCE ( IAIST( 1), FI9820_ST )
EQUIVALENCE ( IAIST( 2), FC9843_ST )
EQUIVALENCE ( IAIST( 3), FCA9812_ST )
EQUIVALENCE ( IAIST( 4), FCA9809_ST )
EQUIVALENCE ( IAIST( 5), FC9873_ST )
EQUIVALENCE ( IAIST( 6), FCA9881_ST )
EQUIVALENCE ( IAIST( 7), AI_9820_1_C4T_ST )
EQUIVALENCE ( IAIST( 8), AI_9820_1_IC4_ST )
EQUIVALENCE ( IAIST( 9), AI_9820_1_NC4_ST )
EQUIVALENCE ( IAIST( 10), AI_9820_1_C3H6_ST )
EQUIVALENCE ( IAIST( 11), AI_9820_1_C3H8_ST )
EQUIVALENCE ( IAIST( 12), AI_9810_1_C3_ST )
EQUIVALENCE ( IAIST( 13), AI_9810_1_IC4_ST )
EQUIVALENCE ( IAIST( 14), AI_9810_1_NC4_ST )
EQUIVALENCE ( IAIST( 15), ALKY_IC42_C3_ST )
EQUIVALENCE ( IAIST( 16), ALKY_IC42_IC4_ST )
EQUIVALENCE ( IAIST( 17), AI_9810_3_IC4_ST )
EQUIVALENCE ( IAIST( 18), TI9808_ST )
EQUIVALENCE ( IAIST( 19), PCA9802_ST )
EQUIVALENCE ( IAIST( 20), TDI9806_ST )
EQUIVALENCE ( IAIST( 21), FC9801_ST )
EQUIVALENCE ( IAIST( 22), FC9805_ST )
EQUIVALENCE ( IAIST( 23), FC9805_VE_ST )
EQUIVALENCE ( IAIST( 24), FC9806_ST )
EQUIVALENCE ( IAIST( 25), FC9836_ST )
EQUIVALENCE ( IAIST( 26), FC9845_ST )
EQUIVALENCE ( IAIST( 27), FC9849_ST )
EQUIVALENCE ( IAIST( 28), FC9852_ST )
EQUIVALENCE ( IAIST( 29), FCA9804_ST )
EQUIVALENCE ( IAIST( 30), FCSA9837_ST )
EQUIVALENCE ( IAIST( 31), FCSA9838_ST )
EQUIVALENCE ( IAIST( 32), TI9865_ST )
EQUIVALENCE ( IAIST( 33), FCA9812_VE_ST )
EQUIVALENCE ( IAIST( 34), FCA9809_VE_ST )
EQUIVALENCE ( IAIST( 35), PDC9804_VE_ST )
C
C
C - equivalence instrument limit minimum
EQUIVALENCE ( LVMI(1), )
C
C
C - equivalence instrument limit maximum
EQUIVALENCE ( LVMA(1), )
C
C
C - equivalence instrument rate of change limit
EQUIVALENCE ( LVCS(1), )
C
C
C - equivalence calculated values
EQUIVALENCE ( CA( 1), ALKYT_IC40LE )
EQUIVALENCE ( CA( 2), ALKYT_IC40LEA )
EQUIVALENCE ( CA( 3), ALKYT_IC40LEB )
EQUIVALENCE ( CA( 4), ALKYT_SYOTIC4 )
EQUIVALENCE ( CA( 5), ALKYT_SYOTC4 )
EQUIVALENCE ( CA( 6), ALKYT_SYOTC3 )
EQUIVALENCE ( CA( 7), ALKYT_SYOTVHT )
EQUIVALENCE ( CA( 8), ALKYT_SYOTPPRA )
EQUIVALENCE ( CA( 9), ALKYT_SYOTNC4 )
Page 11
```

```
ALKYTCALC.txt
EQUIVALENCE ( CA( 10), ALKYT_SYOTC305 )
EQUIVALENCE ( CA( 11), ALKYT_SYOTPRA05 )
EQUIVALENCE ( CA( 12), ALKYT_SYOTC405 )
EQUIVALENCE ( CA( 13), ALKYT_SYOTIC405 )
EQUIVALENCE ( CA( 14), ALKYT_SYOTNC405 )
EQUIVALENCE ( CA( 15), ALKYT_05YHT )
EQUIVALENCE ( CA( 16), ALKYT_HC11 )
EQUIVALENCE ( CA( 17), ALKYT_HC12 )
EQUIVALENCE ( CA( 18), ALKYG_WDOG )
EQUIVALENCE ( CA( 19), ALKYT_D01PCT09 )
EQUIVALENCE ( CA( 20), ALKYT_IC4VAR )
EQUIVALENCE ( CA( 21), ALKYT_TUOTC4 )
EQUIVALENCE ( CA( 22), ALKYT_NC4VAR )
EQUIVALENCE ( CA( 23), ALKYT_C3VAR )
EQUIVALENCE ( CA( 24), ALKYT_TUOTALKY )
EQUIVALENCE ( CA( 25), ALKYT_F980605 )
EQUIVALENCE ( CA( 26), ALKYT_IC4VAR05 )
EQUIVALENCE ( CA( 27), ALKYT_F980905 )
EQUIVALENCE ( CA( 28), ALKYT_C3VAR05 )
EQUIVALENCE ( CA( 29), ALKYT_NC4VAR05 )
EQUIVALENCE ( CA( 30), ALKYT_F983605 )
EQUIVALENCE ( CA( 31), ALKYT_TUOTC405 )
EQUIVALENCE ( CA( 32), ALKYT_TUOTALKY05 )
EQUIVALENCE ( CA( 33), ALKYT_D01RF2DIST )
EQUIVALENCE ( CA( 34), ALKYT_D02RF2DIST )
EQUIVALENCE ( CA( 35), ALKYT_LISAIC4 )
EQUIVALENCE ( CA( 36), ALKYT_IC4F_MY )
EQUIVALENCE ( CA( 37), ALKYT_OLEFF_MY )
EQUIVALENCE ( CA( 38), ALKYT_NC4_MY )
EQUIVALENCE ( CA( 39), ALKYT_ALKY_MY )
EQUIVALENCE ( CA( 40), ALKYT_PRA_MY )
EQUIVALENCE ( CA( 41), ALKYT_D01F_MY )
EQUIVALENCE ( CA( 42), ALKYT_D01_NC4_MY )
EQUIVALENCE ( CA( 43), ALKYT_D01_IC4VAR_MY )
EQUIVALENCE ( CA( 44), ALKYT_D01_IC4_MY )
EQUIVALENCE ( CA( 45), ALKYT_MB_CL )
EQUIVALENCE ( CA( 46), ALKYT_D01MB_CL )
EQUIVALENCE ( CA( 47), ALKYT_PDICVE )
EQUIVALENCE ( CA( 48), ALKYCMI_F9802_WDOG )
EQUIVALENCE ( CA( 49), ALKYCMI_F9804_WDOG )
EQUIVALENCE ( CA( 50), ALKYCMI_F9803_WDOG )

- equivalence calculated value statuses

EQUIVALENCE ( ICAST( 1), ALKYT_IC40LE_ST )
EQUIVALENCE ( ICAST( 2), ALKYT_IC40LEA_ST )
EQUIVALENCE ( ICAST( 3), ALKYT_IC40LEB_ST )
EQUIVALENCE ( ICAST( 4), ALKYT_SYOTIC4_ST )
EQUIVALENCE ( ICAST( 5), ALKYT_SYOTC4_ST )
EQUIVALENCE ( ICAST( 6), ALKYT_SYOTC3_ST )
EQUIVALENCE ( ICAST( 7), ALKYT_SYOTYHT_ST )
EQUIVALENCE ( ICAST( 8), ALKYT_SYOTPRA_ST )
EQUIVALENCE ( ICAST( 9), ALKYT_SYOTNC4_ST )
EQUIVALENCE ( ICAST( 10), ALKYT_SYOTC305_ST )
EQUIVALENCE ( ICAST( 11), ALKYT_SYOTPRA05_ST )
EQUIVALENCE ( ICAST( 12), ALKYT_SYOTC405_ST )
EQUIVALENCE ( ICAST( 13), ALKYT_SYOTIC405_ST )
EQUIVALENCE ( ICAST( 14), ALKYT_SYOTNC405_ST )
EQUIVALENCE ( ICAST( 15), ALKYT_05YHT_ST )
EQUIVALENCE ( ICAST( 16), ALKYT_HC11_ST )
EQUIVALENCE ( ICAST( 17), ALKYT_HC12_ST )
EQUIVALENCE ( ICAST( 18), ALKYG_WDOG_ST )
EQUIVALENCE ( ICAST( 19), ALKYT_D01PCT09_ST )
EQUIVALENCE ( ICAST( 20), ALKYT_IC4VAR_ST )
EQUIVALENCE ( ICAST( 21), ALKYT_TUOTC4_ST )
EQUIVALENCE ( ICAST( 22), ALKYT_NC4VAR_ST )
EQUIVALENCE ( ICAST( 23), ALKYT_C3VAR_ST )
EQUIVALENCE ( ICAST( 24), ALKYT_TUOTALKY_ST )
EQUIVALENCE ( ICAST( 25), ALKYT_F980605_ST )

Page 12
```

```
ALKYTCALC.txt
EQUIVALENCE ( ICAST( 26). ALKYT_IC4VAROS_ST )
EQUIVALENCE ( ICAST( 27). ALKYT_F98090S_ST )
EQUIVALENCE ( ICAST( 28). ALKYT_C3VAROS_ST )
EQUIVALENCE ( ICAST( 29). ALKYT_NC4VAROS_ST )
EQUIVALENCE ( ICAST( 30). ALKYT_F98360S_ST )
EQUIVALENCE ( ICAST( 31). ALKYT_TU0TC40S_ST )
EQUIVALENCE ( ICAST( 32). ALKYT_TU0TALKYOS_ST )
EQUIVALENCE ( ICAST( 33). ALKYT_D01RF2DIST_ST )
EQUIVALENCE ( ICAST( 34). ALKYT_D02RF2DIST_ST )
EQUIVALENCE ( ICAST( 35). ALKYT_LISAIC4_ST )
EQUIVALENCE ( ICAST( 36). ALKYT_IC4F_MY_ST )
EQUIVALENCE ( ICAST( 37). ALKYT_OLEFF_MY_ST )
EQUIVALENCE ( ICAST( 38). ALKYT_NC4_MY_ST )
EQUIVALENCE ( ICAST( 39). ALKYT_ALKY_MY_ST )
EQUIVALENCE ( ICAST( 40). ALKYT_PRA_MY_ST )
EQUIVALENCE ( ICAST( 41). ALKYT_D01F_MY_ST )
EQUIVALENCE ( ICAST( 42). ALKYT_D01_NC4_MY_ST )
EQUIVALENCE ( ICAST( 43). ALKYT_D01_IC4VAR_MY_ST )
EQUIVALENCE ( ICAST( 44). ALKYT_D01_IC4_MY_ST )
EQUIVALENCE ( ICAST( 45). ALKYT_MB_CL_ST )
EQUIVALENCE ( ICAST( 46). ALKYT_D01MB_CL_ST )
EQUIVALENCE ( ICAST( 47). ALKYT_PDICVE_ST )
EQUIVALENCE ( ICAST( 48). ALKYCMI_P9802_WDOG_ST )
EQUIVALENCE ( ICAST( 49). ALKYCMI_F9804_WDOG_ST )
EQUIVALENCE ( ICAST( 50). ALKYCMI_F9803_WDOG_ST )

C
C
C - equivalence floating point manual entry values
C
C EQUIVALENCE ( FME( 1). )
C
C Note! no statuses for manual entries
C
C
C - equivalence controller targets
C
C EQUIVALENCE ( CT( 1). )
C
C - equivalence controller loop ON/OFF states
C
C EQUIVALENCE ( ICT_DBCSS(1). )
C
C
C - equivalence filtered variable values
C
C EQUIVALENCE ( FV( 1). )
C
C - equivalence filtered variable statuses
C
C EQUIVALENCE ( IFVST( 1). )
C
C - equivalence filter factors
C
C EQUIVALENCE ( FF( 1). )
C
C
C - equivalence integer manual entry values
C
C EQUIVALENCE ( IME( 1). ALKYT_CALC_PR )
C EQUIVALENCE ( IME( 2). ALKYT_CALC_LVL )
C EQUIVALENCE ( IME( 3). ALKYT_VALIDCALC )
C
C
C - equivalence digital input value
C
C EQUIVALENCE ( DI( 1). )
C
C - equivalence digital input status
C
C EQUIVALENCE ( IDIST(1). )
C
```

```
ALKYTCALC.txt
C
C - equivalence digital output value, called vdo and not do
C since do is protected
C
C EQUIVALENCE (VDO(1), )
C
C - equivalence digital output status
C
C EQUIVALENCE (IDOST(1), )
C
C-----
C
C - the preprocessor converts these tag names to their database indices
C Note that these data statements can not be commented out, unused
C groups have to be removed
C
C DATA IAI /
C & #FI9820 .
C & #FC9843 .
C & #FCA9812 .
C & #FCA9809 .
C & #FC9873 .
C & #FCA9881 .
C & #AI_9820_1_C4T .
C & #AI_9820_1_IC4 .
C & #AI_9820_1_NC4 .
C & #AI_9820_1_C3H6 .
C & #AI_9820_1_C3H8 .
C & #AI_9810_1_C3 .
C & #AI_9810_1_IC4 .
C & #AI_9810_1_NC4 .
C & #ALKY_IC42_C3 .
C & #ALKY_IC42_IC4 .
C & #AI_9810_5_IC4 .
C & #TI9808 .
C & #PCA9802 .
C & #TDI9806 .
C & #FC9801 .
C & #FC9805 .
C & #FC9805_VE .
C & #FC9806 .
C & #FC9836 .
C & #FC9845 .
C & #FC9849 .
C & #FC9852 .
C & #FCA9804 .
C & #FCSA9837 .
C & #FCSA9838 .
C & #TI9865 .
C & #FCA9812_VE .
C & #FCA9809_VE .
C & #PDC9804_VE /
C
C DATA ILV //
C
C DATA ICA /
C & #ALKYT_IC40LE .
C & #ALKYT_IC40LEA .
C & #ALKYT_IC40LEB .
C & #ALKYT_SYOTIC4 .
C & #ALKYT_SYOTC4 .
C & #ALKYT_SYOTC3 .
C & #ALKYT_SYOTYHT .
C & #ALKYT_SYOTPR4 .
C & #ALKYT_SYOTNC4 .
C
```

```

C      &          #ALKYT_SYOTC30S      ALKYTCALC.txt      .
C      &          #ALKYT_SYOTPRA0S     .
C      &          #ALKYT_SYOTC40S     .
C      &          #ALKYT_SYOTIC40S     .
C      &          #ALKYT_SYOTNC40S     .
C      &          #ALKYT_05VHT         .
C      &          #ALKYT_HCL1          .
C      &          #ALKYT_HCL2          .
C      &          #ALKYG_WDOG           .
C      &          #ALKYT_D01PCT09      .
C      &          #ALKYT_IC4VAR         .
C      &          #ALKYT_TU0TC4         .
C      &          #ALKYT_NC4VAR         .
C      &          #ALKYT_C3VAR         .
C      &          #ALKYT_TUOTALKY       .
C      &          #ALKYT_F98060S       .
C      &          #ALKYT_IC4VAR0S      .
C      &          #ALKYT_F98090S      .
C      &          #ALKYT_C3VAR0S      .
C      &          #ALKYT_NC4VAR0S      .
C      &          #ALKYT_F98360S      .
C      &          #ALKYT_TU0TC40S     .
C      &          #ALKYT_TUOTALKY0S    .
C      &          #ALKYT_D01RF2DIST    .
C      &          #ALKYT_D02RF2DIST    .
C      &          #ALKYT_LISAIC4       .
C      &          #ALKYT_IC4F_MY       .
C      &          #ALKYT_0LEFF_MY      .
C      &          #ALKYT_NC4_MY        .
C      &          #ALKYT_ALKY_MY        .
C      &          #ALKYT_PRA_MY        .
C      &          #ALKYT_D01F_MY       .
C      &          #ALKYT_D01_NC4_MY    .
C      &          #ALKYT_D01_IC4VAR_MY .
C      &          #ALKYT_D01_IC4_MY   .
C      &          #ALKYT_MB_CL        .
C      &          #ALKYT_D01MB_CL     .
C      &          #ALKYT_PDICVE       .
C      &          #ALKYCM1_P9802_WDOG  .
C      &          #ALKYCM1_F9804_WDOG .
C      &          #ALKYCM1_F9803_WDOG /

CC      DATA IFME //

CC      DATA ICT //

CC      DATA IFV //

C      DATA IIME /
C      &          #ALKYT_CALC_PR      .
C      &          #ALKYT_CALC_LVL     .
C      &          #ALKYT_VALIDCALC    /

CC      DATA IDI //

CC      DATA IDO //

C
C
C.....
C.....
C.....

```

```

C
C                                     ALKYTCALC.txt
C
C$INPUT
C
C=====
C
C   - Set Filtering Module Initialization Flag "ON" For First Iteration
C
C   INIEXI = 0
C
C.....
C
C   -Return to Here After Waking from Hibernation
C
C 1 CONTINUE
C
C.....
C
C   - Read the Integer manual flags for debug printing
C
C   IFERR = 10
C   CALL DBGETI ( IIMEMX ,           ! number of variables
C &             IIME  ,           ! table for Tagnames of variables
C &             IIME  ,           ! table for values of variables
C &             IERR  )           ! error return code
C   IF ( IERR .NE. 0 ) GOTO $100
C
C=====
C
C   - Open the DEBUG PRINT File
C
C-----
C
C   JPRINT = 0
C   JPRNT  = 0
C   IDETAL = 0
C   ILUNUM = 3
C
C   IF ( ALKYT_CALC_PR .NE. 0 ) THEN
C     JPRINT = 1
C     OPEN (UNIT=ILUNUM, FILE='ALKY_RUN:ALKYTCALC.RUN', STATUS='NEW')
C     IDETAL = ALKYT_CALC_LVL
C     JDETAL = IDETAL - 1
C   ENDIF
C
C   - print the DEBUG file header
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL DATE_AND_TIME( CDATE,CTIME )
C
C     CALL PLABEL( '+++++' )
C     CALL PLABEL( '          START OF ALKY CALCULATIONS ' )
C     WRITE(ILUNUM, 9003) CDATE, CTIME
C     CALL PLABEL( '+++++' )
C   ENDIF
C
C   JPRINT = JPRINT
C   IF ( JPRINT .GT. 0 ) THEN
C     JPRNT = ILUNUM
C   ENDIF
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL ( ' ' )
C     CALL PLABEL ( ' The debug print files is open' )
C     CALL PLABEL ( ' ' )
C   ENDIF
C
C=====
C
C   - read the analog inputs

```

```

C
C                                     ALKYTCALC.txt
C
IFERR = 20
CALL DBGETF ( IAIMX ,           ! number of variables
&           IAI   ,           ! table for Tagnames of variables
&           AI    ,           ! table for values of variables
&           IAIST ,           ! table for status of variables
&           IERR )           ! error return code
IF ( IERR .NE. 0 ) GOTO $100
C
IF ( JPRINT .GT. 0 ) THEN
  CALL LABEL ( ' ' )
  CALL LABEL ( ' Gone through AI read ' )
ENDIF
C
C.....
C
C - read the floating point manual entries
C
IFERR = 30
CALL DBGETF ( IFMEMX ,         ! number of variables
&           IFME ,           ! table for Tagnames of variables
&           FME   ,           ! table for values of variables
&           IFMEST ,         ! table for status of variables
&           IERR )           ! error return code
IF ( IERR .NE. 0 ) GOTO $100
C
IF ( JPRINT .GT. 0 ) THEN
  CALL LABEL ( ' ' )
  CALL LABEL ( ' Gone through ME read ' )
ENDIF
C
C.....
C
C - read the calculated variables
C
IFERR = 40
CALL DBGETF ( ICAMX ,         ! number of variables
&           ICA   ,           ! table for Tagnames of variables
&           CA    ,           ! table for values of variables
&           ICAST ,         ! table for status of variables
&           IERR )           ! error return code
IF ( IERR .NE. 0 ) GOTO $100
C
IF ( JPRINT .GT. 0 ) THEN
  CALL LABEL ( ' ' )
  CALL LABEL ( ' Gone through CA read ' )
ENDIF
C
C.....
C
C - read controller targets
C
IFERR = 50
CALL OICONGET(ICTMX,
&           ICT,           ! control loop index
&           CT,           ! target value
&           ICT_DBCSS,    ! control loop state
&           ICT_PCTP,    ! control loop test output
&           RCT_PCDM,    ! Delta move
&           RCT_PCPE,    ! Previous error
&           RCT_PCPE,    ! Prior to previous error
&           RCT_PCPCV,   ! Control var. n-1
&           RCT_PCPCV,   ! Control var. n-2
&           RCT_DBVMIN , ! target minimum
&           RCT_DBVMAX , ! target maximum
&           IERR )       ! error return code
IF ( IERR .NE. 0 ) GOTO $100
C
IF ( JPRINT .GT. 0 ) THEN

```

```

CC          ALKYTCALC.txt
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( ' Gone through controller read ' )
CC      ENDIF
C
C .....
C
C      - read the filtered calculated variables
C
CC      IFERR = 60
CC      CALL DBGETF ( IFVMX ,      ! number of variables
CC      &          IFV ,          ! table for Tagnames of variables
CC      &          FV ,          ! table for values of variables
CC      &          IFVST ,       ! table for status of variables
CC      &          IERR )       ! error return code
CC      IF ( IERR .NE. 0 ) GOTO $100
C
CC      IF ( JPRINT .GT. 0 ) THEN
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( ' Gone through all filtered variable read ' )
CC      ENDIF
C
C .....
C
C      - read the filter factors, ( FF(I)'s )
C
CC      IFERR = 70
CC      CALL OIGETFF(IFVMX ,      ! number of variables
CC      &          IFV ,          ! tagname of variables
CC      &          FF ,          ! filter factors
CC      &          DBFE ,       ! filter information
CC      &          IERR )       ! error return code
CC      IF ( IERR .NE. 0 ) GOTO $100
C
CC      IF ( JPRINT .GT. 0 ) THEN
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( ' Gone through filter factor read ' )
CC      ENDIF
C
C .....
C
C      - read the digital inputs
C
CC      IFERR = 100
C
CC      CALL DBGETD      ( IDIMX ,      ! number of variables
CC      &          IDI ,      ! indices database variables
CC      &          DI ,      ! value of digital input
CC      &          IERR )   ! error return code
C
CC      IF ( IERR .NE. 0 ) GOTO $100
CC      IF ( JPRINT .GT. 0 ) THEN
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( ' Gone through digital input reading ' )
CC      ENDIF
C
CC      DO I = 1, IDIMX
CC      IF ( DI(I) .EQ. 0 ) THEN
CC      IDIST(I) = BAD_STATUS
CC      VALUE = FALSE_VALUE ! GIVE ANY VALUE TO VALUE
CC      ELSE IF ( DI(I) .EQ. 1 ) THEN
CC      VALUE = FALSE_VALUE
CC      IDIST(I) = GOOD_STATUS
CC      ELSE
CC      VALUE = TRUE_VALUE
CC      IDIST(I) = GOOD_STATUS
CC      END IF
CC      DI(I) = VALUE

```

```

                                ALKYTCALC.txt
CC      ENDDO
C
IF ( JPRINT .GT. 0 ) THEN
  CALL LABEL ( ' ' )
  CALL LABEL ( ' Gone through all standard reading ' )
ENDIF

C-----
C=====
C
C$DEBUG PRINT
C      - Print the Database Inputs
C
IF ( JPRINT .GT. 0 ) THEN
  CALL LABEL ( ' ' )
  CALL LABEL ( ' PROGRAM INPUTS ' )
  CALL LABEL ( ' ' )
C
  CALL LABEL ( ' ' )
  CALL LABEL ( '***** INSTRUMENTS' )
  IFERR = 160
  CALL DBTEXT ( IAIMX, IAI, %REF(AITEXT), IERR )
  IF ( IERR .NE. 0 ) THEN
    CALL MESSAGES
    &      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
  &
  ENDDO
  DO I = 1, IAIMX
    WRITE(ILUNUM, 9000) AITEXT(I)( 1:20), AITEXT(I)(21:52),
    &      AI(I), AITEXT(I)(53:60), IAIST(I)
  &
  ENDDO
C
  CALL LABEL ( ' ' )
  CALL LABEL ( '***** FLOATING POINT MANUAL ENTRIES' )
  IFERR = 170
  CALL DBTEXT ( IFMEMX, IFME, %REF(FMETEXT), IERR )
  IF ( IERR .NE. 0 ) THEN
    CALL MESSAGES
    &      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
  &
  ENDDO
  DO I = 1, IFMEMX
    WRITE(ILUNUM, 9000) FMETEXT(I)( 1:20), FMETEXT(I)(21:52),
    &      FME(I), FMETEXT(I)(53:60), IFMEST(I)
  &
  ENDDO
C
  CALL LABEL ( ' ' )
  CALL LABEL ( '***** INTEGER MANUAL ENTRIES' )
  IFERR = 180
  CALL DBTEXT ( IIMEMX, IIME, %REF(IMETEXT), IERR )
  IF ( IERR .NE. 0 ) THEN
    CALL MESSAGES
    &      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
  &
  ENDDO
  DO I = 1, IIMEMX
    WRITE(ILUNUM, 9000) IMETEXT(I)( 1:20), IMETEXT(I)(21:52),
    &      FLOAT(IME(I)), IMETEXT(I)(53:60), 1
  &
  ENDDO
C
  CALL LABEL ( ' ' )
  CALL LABEL ( '***** OLD FILTERED VALUES, FILTER FACTORS AND'//
  &      ' FILTERING ENABLE BIT (MUST BE 0)' )
  IFERR = 190
  CALL DBTEXT ( IFVMX, IFV, %REF(FVTEXT), IERR )
  IF ( IERR .NE. 0 ) THEN
    CALL MESSAGES
    &      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
  &
  ENDDO
  DO I = 1, IFVMX

```

```

CC      ALKYTCALC.txt
CC      WRITE(ILUNUM, 9005) FVTEXT(I)( 1:20), FVTEXT(I)(21:52),
CC      & FV(I), 'FACTOR = ', FF(I), DBFE(I)
CC      ENDDO
CC
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( '***** MIN LIMIT VALUES ' )
CC      IFERR = 200
CC      CALL DBTEXT ( ILVMX, ILV, %REF(LVMITEXT), IERR )
CC      IF ( IERR .NE. 0 ) THEN
CC      CALL MESSAGES
CC      & (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
CC      ENDDO
CC      ENDDO
CC      DO I = 1, ILVMX
CC      WRITE(ILUNUM, 9000) LVMITEXT(I)( 1:20), LVMITEXT(I)(21:52),
CC      & LVMI(I), LVMITEXT(I)(53:60), 1
CC      ENDDO
CC
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( '***** MAX LIMIT VALUES ' )
CC      IFERR = 210
CC      CALL DBTEXT ( ILVMX, ILV, %REF(LVMATEXT), IERR )
CC      IF ( IERR .NE. 0 ) THEN
CC      CALL MESSAGES
CC      & (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
CC      ENDDO
CC      ENDDO
CC      DO I = 1, ILVMX
CC      WRITE(ILUNUM, 9000) LVMATEXT(I)( 1:20), LVMATEXT(I)(21:52),
CC      & LVMA(I), LVMATEXT(I)(53:60), 1
CC      ENDDO
CC
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( '***** MAX ROC VALUES ' )
CC      IFERR = 215
CC      CALL DBTEXT ( ILVMX, ILV, %REF(LVCSTEXT), IERR )
CC      IF ( IERR .NE. 0 ) THEN
CC      CALL MESSAGES
CC      & (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
CC      ENDDO
CC      ENDDO
CC      DO I = 1, ILVMX
CC      WRITE(ILUNUM, 9000) LVCSTEXT(I)( 1:20), LVCSTEXT(I)(21:52),
CC      & LVCS(I), LVCSTEXT(I)(53:60), 1
CC      ENDDO
CC
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( '***** DIGITAL INPUTS' )
CC      IFERR = 220
CC      CALL DBTEXT ( IDIMX, IDI, %REF(DITEXT), IERR )
CC      IF ( IERR .NE. 0 ) THEN
CC      CALL MESSAGES
CC      & (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
CC      ENDDO
CC      ENDDO
CC      DO I = 1, IDIMX
CC      WRITE(ILUNUM, 9000) DITEXT(I)( 1:20), DITEXT(I)(21:52),
CC      & FLOAT(DI(I)), DITEXT(I)(53:60), IDIST(I)
CC      ENDDO
CC
CC      CALL PLABEL ( ' ' )
CC      CALL PLABEL ( '***** CONTROLLER TARGETS' )
CC      IFERR = 230
CC      CALL DBTEXT ( ICTMX, ICT, %REF(CTTEXT), IERR )
CC      IF ( IERR .NE. 0 ) THEN
CC      CALL MESSAGES
CC      & (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
CC      ENDDO
CC      ENDDO
CC      DO I = 1, ICTMX
CC      WRITE(ILUNUM, 9002) CTTEXT(I)( 1:20), CTTEXT(I)(21:52),
CC      & CT(I), ICT_DBCSS(I)
CC      ENDDO

```



```

                                ALKYTCALC.txt
&      + ( VNC4      ° DNNC4      ) + ( VC3H6      ° DNC3H6 )
&      + ( VC3H8      ° DNC3H8      )
C
      MC4T      = VC4T      ° DNC4T      / MSUM
      MIC4      = VIC4      ° DNIC4      / MSUM
      MNC4      = VNC4      ° DNNC4      / MSUM
      MC3H6     = VC3H6     ° DNC3H6     / MSUM
      MC3H8     = VC3H8     ° DNC3H8     / MSUM
C
      MC4T_ST   = GOOD_STATUS
      MIC4_ST   = GOOD_STATUS
      MNC4_ST   = GOOD_STATUS
      MC3H6_ST  = GOOD_STATUS
      MC3H8_ST  = GOOD_STATUS
      VC4T_ST   = GOOD_STATUS
C
      ELSE
C
      MC4T_ST   = BAD_STATUS
      MIC4_ST   = BAD_STATUS
      MNC4_ST   = BAD_STATUS
      MC3H6_ST  = BAD_STATUS
      MC3H8_ST  = BAD_STATUS
      VC4T_ST   = BAD_STATUS
C
      ENDIF
C
      .....
C
      IF ( JPRINT .GT. 0 ) THEN
C
      CALL PLABEL( ' ' )
      CALL PLABEL( 'The variables needed for calculations' )
      WRITE ( ILUNUM, 9040 ) 'VSUM (m3/kmol)' . . . VSUM
      WRITE ( ILUNUM, 9040 ) 'VC4T (-)' . . . VC4T
      WRITE ( ILUNUM, 9040 ) 'VIC4 (-)' . . . VIC4
      WRITE ( ILUNUM, 9040 ) 'VNC4 (-)' . . . VNC4
      WRITE ( ILUNUM, 9040 ) 'VC3H6 (-)' . . . VC3H6
      WRITE ( ILUNUM, 9040 ) 'VC3H8 (-)' . . . VC3H8
      WRITE ( ILUNUM, 9040 ) 'MSUM (kg/m3)' . . . MSUM
      WRITE ( ILUNUM, 9040 ) 'MC4T (-)' . . . MC4T
      WRITE ( ILUNUM, 9040 ) 'MIC4 (-)' . . . MIC4
      WRITE ( ILUNUM, 9040 ) 'MNC4 (-)' . . . MNC4
      WRITE ( ILUNUM, 9040 ) 'MC3H6 (-)' . . . MC3H6
      WRITE ( ILUNUM, 9040 ) 'MC3H8 (-)' . . . MC3H8
      CALL PLABEL( ' ' )
C
      ENDIF
C
      .....
C
      - calculate the density of the olefin feed (kg/m3)
C
      ISTATUS = MC4T_ST ° MIC4_ST ° MNC4_ST
      &        ° MC3H6_ST ° MC3H8_ST
C
      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
      DEN = ( MC4T      ° DNC4T      ) + ( MIC4      ° DNIC4      )
      &      + ( MNC4      ° DNNC4      )
      &      + ( MC3H6     ° DNC3H6     ) + ( MC3H8     ° DNC3H8     )
C
      DEN_ST = GOOD_STATUS
      ELSE
      DEN_ST = BAD_STATUS
      ENDIF
C
      IF ( JPRINT .GT. 0 ) THEN
      CALL PLABEL( ' ' )

```

```

                                ALKYTCALC.txt
CALL PLABEL('The density of the flow FIC-9812')
WRITE ( ILUNUM, 9040 ) 'DEN (kg/m3)' . 'i', DEN
CALL PLABEL('')
ENDIF
C
C-----
C
C - calculate the total olefin volume flow of FIC-9812 (m3/h)
C
C ISTATUS = FCA9812_ST * DEN_ST * VC4T_ST
C
C IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     VOLEF = FCA9812 / DEN * ( VC4T + VC3H6 ) * 1000
C
C     VOLEF_ST = GOOD_STATUS
C ELSE
C     VOLEF_ST = BAD_STATUS
C ENDIF
C
C-----
C
C I-A-2 The isobutane makeup
C
C - ISOBUTANE MAKEUP ( measurements: FCA9809, AI_9810_1_IC4,
C AI_9810_1_NC4, AI_9810_1_C3 )
C
C IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL('')
C     CALL PLABEL('I-A-2 The isobutane makeup')
C     CALL PLABEL('')
C ENDIF
C
C ISTATUS = AI_9810_1_IC4_ST * AI_9810_1_NC4_ST
C & AI_9810_1_C3_ST
C
C IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C - change the mole percentage values to the volume fractions
C
C     VSUM9809 = ( ( AI_9810_1_IC4 / 100 * MWIC4 ) / DNIC4 )
C & + ( ( AI_9810_1_NC4 / 100 * MWNC4 ) / DNNC4 )
C & + ( ( AI_9810_1_C3 / 100 * MWC3H8 ) / DNC3H8 )
C
C     V9809IC4 = ( AI_9810_1_IC4 / 100 * MWIC4 ) / DNIC4
C & / VSUM9809
C     V9809NC4 = ( AI_9810_1_NC4 / 100 * MWNC4 ) / DNNC4
C & / VSUM9809
C     V9809C3H8 = ( AI_9810_1_C3 / 100 * MWC3H8 ) / DNC3H8
C & / VSUM9809
C
C - change the volume ratios to the mass ratios
C
C     MSUM9809 = ( V9809IC4 * DNIC4 )
C & + ( V9809NC4 * DNNC4 )
C & + ( V9809C3H8 * DNC3H8 )
C
C     M9809IC4 = V9809IC4 * DNIC4 / MSUM9809
C     M9809NC4 = V9809NC4 * DNNC4 / MSUM9809
C     M9809C3H8 = V9809C3H8 * DNC3H8 / MSUM9809
C
C     M9809IC4_ST = GOOD_STATUS
C     M9809NC4_ST = GOOD_STATUS
C     M9809C3H8_ST = GOOD_STATUS
C
C ELSE
C

```

```

                                ALKYTCALC.txt
M9809IC4_ST = BAD_STATUS
M9809NC4_ST = BAD_STATUS
M9809C3H8_ST = BAD_STATUS
C
C   ENDIF
C
C-----
C   IF ( JPRINT .GT. 0 ) THEN
C
C     CALL PLABEL( ' ' )
C     CALL PLABEL( 'The variables needed for calculations' )
C     WRITE ( ILUNUM, 9040 ) 'VSUM9809 (m3/kmol) ' , ' ', VSUM9809
C     WRITE ( ILUNUM, 9040 ) 'V9809IC4 (-) ' , ' ', V9809IC4
C     WRITE ( ILUNUM, 9040 ) 'V9809NC4 (-) ' , ' ', V9809NC4
C     WRITE ( ILUNUM, 9040 ) 'V9809C3H8 (-) ' , ' ', V9809C3H8
C     WRITE ( ILUNUM, 9040 ) 'MSUM9809 (kg/m3) ' , ' ', MSUM9809
C     WRITE ( ILUNUM, 9040 ) 'M9809IC4 (-) ' , ' ', M9809IC4
C     WRITE ( ILUNUM, 9040 ) 'M9809NC4 (-) ' , ' ', M9809NC4
C     WRITE ( ILUNUM, 9040 ) 'M9809C3H8 (-) ' , ' ', M9809C3H8
C     CALL PLABEL( ' ' )
C
C   ENDIF
C
C-----
C   - calculate the density of isobutane makeup (kg/m3)
C
C   ISTATUS = M9809IC4_ST * M9809NC4_ST * M9809C3H8_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     DEN9809 = ( M9809IC4 * DNIC4 )
C     &         + ( M9809NC4 * DNNC4 )
C     &         + ( M9809C3H8 * DNC3H8 )
C
C     DEN9809_ST = GOOD_STATUS
C   ELSE
C     DEN9809_ST = BAD_STATUS
C   ENDIF
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL( ' ' )
C     CALL PLABEL( 'The density of the flow FIC-9809' )
C     WRITE ( ILUNUM, 9040 ) 'DEN9809 (kg/m3) ' , ' ', DEN9809
C     CALL PLABEL( ' ' )
C   ENDIF
C
C-----
C   I-B The total feed of hydrocarbons to the reactor
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL( ' ' )
C     CALL PLABEL( 'I-B The total feed of hydrocarbons ' )
C     CALL PLABEL( 'to the reactor' )
C     CALL PLABEL( ' ' )
C   ENDIF
C
C   - calculate the mass flow of propene (t/h) in the feed to the alky
C
C   ISTATUS = FCA9812_ST * MC3H6_ST * FCA9812_VE_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C     IF ( ( FCA9812_VE .GT. 0.0 ) .AND.
C     &     ( MC3H6 .GE. 0.0 ) ) THEN
C

```

```

                                ALKYTCALC.txt
      ALKYT_SYOTC3 = FCA9812 * MC3H6
    ELSE
      ALKYT_SYOTC3 = 0.0
    ENDIF
C
      ALKYT_SYOTC3_ST = GOOD_STATUS
    ELSE
      ALKYT_SYOTC3_ST = BAD_STATUS
    ENDIF
C
C.....
C
C - calculate the mass flow of c4-olefin in the feed to the alky
C
      ISTATUS = FCA9812_ST * MC4T_ST * FCA9812_VE_ST
C
      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
        IF ( FCA9812_VE .GT. 0.0 ) THEN
C
          ALKYT_SYOTC4 = FCA9812 * MC4T
        ELSE
          ALKYT_SYOTC4 = 0.0
        ENDIF
C
        ALKYT_SYOTC4_ST = GOOD_STATUS
      ELSE
        ALKYT_SYOTC4_ST = BAD_STATUS
      ENDIF
C
C.....
C
C - calculate the mass flow of the isobutane fresh feed to the alky (t/h)
C
      ISTATUS = FCA9812_ST * FCA9809_ST * M9809IC4_ST
      &          * MIC4_ST * FCA9812_VE_ST * FCA9809_VE_ST
C
      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
        IF (( FCA9812_VE .GT. 0.0 ) .OR.
          &    ( FCA9809_VE .GT. 0.0 )) THEN
C
          ALKYT_SYOTIC4 = FCA9809 * M9809IC4 + FCA9812 * MIC4
        ELSE
          ALKYT_SYOTIC4 = 0.0
        ENDIF
C
        ALKYT_SYOTIC4_ST = GOOD_STATUS
      ELSE
        ALKYT_SYOTIC4_ST = BAD_STATUS
      ENDIF
C
C.....
C
C - calculate the amount of n-butane in the feed to the alky (t/h)
C
      ISTATUS = FCA9809_ST * M9809NC4_ST * FCA9812_ST * MNC4_ST
      &          * FCA9812_VE_ST * FCA9809_VE_ST
C
      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
        IF (( FCA9812_VE .GT. 0.0 ) .OR.
          &    ( FCA9809_VE .GT. 0.0 )) THEN
C
          ALKYT_SYOTNC4 = FCA9809 * M9809NC4
          &                + FCA9812 * MNC4
        ELSE
          ALKYT_SYOTNC4 = 0.0
        ENDIF

```

```

C                                     ALKYTCALC.txt
C
C     ALKYT_SYOTNC4_ST = GOOD_STATUS
C ELSE
C     ALKYT_SYOTNC4_ST = BAD_STATUS
C ENDIF
C
C .....
C
C - calculate the amount of propane in the feed to the alky (t/h)
C from the analysers information
C
C ISTATUS = FCA9809_ST * M9809C3H8_ST
C &         * FCA9812_ST * MC3H8_ST
C &         * FCA9812_VE_ST * FCA9809_VE_ST
C
C IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     IF (( FCA9812_VE .GT. 0.0 ) .OR.
C &       ( FCA9809_VE .GT. 0.0 )) THEN
C
C         ALKYT_SYOTPRA = FCA9809 * M9809C3H8
C &                   + FCA9812 * MC3H8
C     ELSE
C         ALKYT_SYOTPRA = 0.0
C     ENDIF
C
C     ALKYT_SYOTPRA_ST = GOOD_STATUS
C ELSE
C     ALKYT_SYOTPRA_ST = BAD_STATUS
C ENDIF
C
C .....
C
C - the total feed to the alky according to analysers information
C
C ISTATUS = ALKYT_SYOTC3_ST * ALKYT_SYOTC4_ST * ALKYT_SYOTIC4_ST
C &         * ALKYT_SYOTNC4_ST * ALKYT_SYOTPRA_ST
C
C IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     ALKYT_SYOTYHT = ALKYT_SYOTC3 + ALKYT_SYOTC4
C &                 + ALKYT_SYOTIC4
C &                 + ALKYT_SYOTNC4 + ALKYT_SYOTPRA
C
C     ALKYT_SYOTYHT_ST = GOOD_STATUS
C ELSE
C     ALKYT_SYOTYHT_ST = BAD_STATUS
C ENDIF
C
C .....
C
C - calculate the percentage values in the feed
C
C ISTATUS = ALKYT_SYOTYHT_ST
C
C IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     IF ( ALKYT_SYOTYHT .GT. 1.0 ) THEN
C
C         ALKYT_SYOTC40S = ALKYT_SYOTC4 * 100 / ALKYT_SYOTYHT
C         ALKYT_SYOTIC40S = ALKYT_SYOTIC4 * 100 / ALKYT_SYOTYHT
C         ALKYT_SYOTNC40S = ALKYT_SYOTNC4 * 100 / ALKYT_SYOTYHT
C         ALKYT_SYOTC30S = ALKYT_SYOTC3 * 100 / ALKYT_SYOTYHT
C         ALKYT_SYOTPRA0S = ALKYT_SYOTPRA * 100 / ALKYT_SYOTYHT
C
C     ELSE
C
C         ALKYT_SYOTC40S = 0
C         ALKYT_SYOTIC40S = 0

```

```

                                ALKYTCALC.txt
    ALKYT_SYOTNC405 = 0
    ALKYT_SYOTC305 = 0
    ALKYT_SYOTPRA05 = 0
C
C   ENDIF
C
    ALKYT_SYOTC405_ST = GOOD_STATUS
    ALKYT_SYOTIC405_ST = GOOD_STATUS
    ALKYT_SYOTNC405_ST = GOOD_STATUS
    ALKYT_SYOTC305_ST = GOOD_STATUS
    ALKYT_SYOTPRA05_ST = GOOD_STATUS
C
C   ELSE
C
    ALKYT_SYOTC405_ST = BAD_STATUS
    ALKYT_SYOTIC405_ST = BAD_STATUS
    ALKYT_SYOTNC405_ST = BAD_STATUS
    ALKYT_SYOTC305_ST = BAD_STATUS
    ALKYT_SYOTPRA05_ST = BAD_STATUS
C
C   ENDIF
C
C
C   ISTATUS = ALKYT_SYOTC405_ST * ALKYT_SYOTIC405_ST
& * ALKYT_SYOTNC405_ST
& * ALKYT_SYOTC305_ST * ALKYT_SYOTPRA05_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
    ALKYT_OSYHT = ALKYT_SYOTC405 + ALKYT_SYOTIC405
& + ALKYT_SYOTNC405
& + ALKYT_SYOTC305 + ALKYT_SYOTPRA05
C
    ALKYT_OSYHT_ST = GOOD_STATUS
    ELSE
    ALKYT_OSYHT_ST = BAD_STATUS
    ENDIF
C
C-----
C
C   I-B-1 The olefin feed to the reactor
C
    IF ( JPRINT .GT. 0 ) THEN
        CALL PLABEL( ' ' )
        CALL PLABEL('I-B-1 The olefin feed to the reactor,')
        CALL PLABEL(' volume flow')
        CALL PLABEL( ' ' )
    ENDIF
C
C - calculate the volume flows of the fresh olefin (m3/h) to the upper
C nozzles GD-9801A and to the lower nozzles GD-9801B
C
    ISTATUS = VOLEF_ST * FCA9812_ST * FC9873_ST * FCA9809_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
    NUMER = FC9873 * VOLEF
    DENOM = FCA9812 + FCA9809
C
    VOLEFA = DIVCHK ( JPRNT, JDETAL,
& NUMER, DENOM, IERROR)
C
    ENDIF
C
C   IF (( ISTATUS .NE. BAD_STATUS ) .AND.
& ( IERROR .EQ. GOOD_IER )) THEN
C
    VOLEFB = VOLEF - VOLEFA

```

```

C                                     ALKYTCALC.txt
C
C      VOLEFA_ST = GOOD_STATUS
C      VOLEFB_ST = GOOD_STATUS
C      ELSE
C      VOLEFA_ST = BAD_STATUS
C      VOLEFB_ST = BAD_STATUS
C      ENDIF
C
C      IF ( JPRINT .GT. 0 ) THEN
C
C      CALL PLABEL( ' ' )
C      CALL PLABEL('The olefin fresh feed to the alky')
C      CALL PLABEL('The total olefin volume flow')
C      WRITE ( ILUNUM, 9040 ) 'VOLEF (m3/h)' ., VOLEF
C      CALL PLABEL('The volume flows of the fresh olefin to')
C      CALL PLABEL('the upper nozzles GD-9801A')
C      WRITE ( ILUNUM, 9040 ) 'VOLEFA (m3/h)' ., VOLEFA
C      CALL PLABEL('The volume flows of the fresh olefin to')
C      CALL PLABEL('the lower nozzles GD-9801B')
C      WRITE ( ILUNUM, 9040 ) 'VOLEFB (m3/h)' ., VOLEFB
C      CALL PLABEL( ' ' )
C
C      ENDIF
C
C-----
C
C      I-B-2 The molecular weights of the flows FIC-9809, FIC-9812,
C      FI-9820 and FIC-9843
C
C      IF ( JPRINT .GT. 0 ) THEN
C      CALL PLABEL( ' ' )
C      CALL PLABEL('I-B-2 The molecular weights of the flows')
C      CALL PLABEL(' FIC-9809, FIC-9812, FI-9820')
C      CALL PLABEL(' and FIC-9843')
C      ENDIF
C
C      - calculate the molecular weights of the flows (kg/kmo1)
C
C      ISTATUS = AI_9810_I_C3_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C      MW9809 = ( AI_9810_I_C3 / 100 * MWC3H8 ) ! The isobutane makeup
C      & + ((1 - AI_9810_I_C3 / 100 ) * MWIC4 ) ! (FCA9809)
C
C      MW9809_ST = GOOD_STATUS
C      ELSE
C      MW9809_ST = BAD_STATUS
C      ENDIF
C
C
C      ISTATUS = AI_9820_I_C4T_ST * AI_9820_I_IC4_ST
C      & * AI_9820_I_C3H8_ST * AI_9820_I_NC4_ST
C      & * AI_9820_I_C3H6_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C      MW9812 = ( AI_9820_I_C4T / 100 * MWC4T ) ! The olefin feed
C      & + ( AI_9820_I_IC4 / 100 * MWIC4 ) ! (FCA9812)
C      & + ( AI_9820_I_C3H8 / 100 * MWC3H8 )
C      & + ( AI_9820_I_NC4 / 100 * MWNC4 )
C      & + ( AI_9820_I_C3H6 / 100 * MWC3H6 )
C
C      MW9812_ST = GOOD_STATUS
C      ELSE
C      MW9812_ST = BAD_STATUS
C      ENDIF
C

```

```

C                                     ALKYTCALC.txt
C
C   ISTATUS = ALKY_IC42_C3_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     MW9820 = ( ALKY_IC42_C3 / 100 * MWC3H8 ) ! The isobutane recycle
C   &      + ((1 - ALKY_IC42_C3 / 100 ) * MWIC4 ) ! from DA-9802
C                                           ! (FI9820)
C
C     MW9820_ST = GOOD_STATUS
C   ELSE
C     MW9820_ST = BAD_STATUS
C   ENDIF
C
C
C   - ALKY_IC43_IC4M is a value of laboratory test and its value is
C     approximately 85 mass-%. This test value is replaced with a
C     variable F9843_IC4M, which is given a constant value 85 mass-%.
C     Change the value to mole-%.
C
C     MPIC43 = F9843_IC4M / 100 * MWIC4
C   &      / ( F9843_IC4M / 100 * MWIC4
C   &      + ( 100 - F9843_IC4M ) / 100 * MWC3H8 )
C
C     MW9843 = MPIC43 * MWIC4 + ( 1 - MPIC43 ) * MWC3H8 ! The isobutane
C                                                         ! recycle from
C                                                         ! DA-9803 (FC9843)
C
C   IF ( JPRINT .GT. 0 ) THEN
C
C     CALL LABEL( ' ' )
C     CALL LABEL('The molecular weights of the flows')
C     WRITE ( ILUNUM, 9040 ) 'MW9809 (kg/kmol)' , ':', MW9809
C     WRITE ( ILUNUM, 9040 ) 'MW9812 (kg/kmol)' , ':', MW9812
C     WRITE ( ILUNUM, 9040 ) 'MW9820 (kg/kmol)' , ':', MW9820
C     WRITE ( ILUNUM, 9040 ) 'MW9843 (kg/kmol)' , ':', MW9843
C     CALL LABEL('The variable needed for calculations')
C     CALL LABEL('The isobutane mole-% of flow FC9843')
C     WRITE ( ILUNUM, 9040 ) 'MPIC43 (-)' , ':', MPIC43
C     CALL LABEL( ' ' )
C
C   ENDIF
C
C-----+-----
C
C   I-B-3 The isobutane feed to the reactor, mole flows
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL LABEL( ' ' )
C     CALL LABEL('I-B-3 The isobutane feed to the reactor,')
C     CALL LABEL(' mole flows')
C     CALL LABEL( ' ' )
C   ENDIF
C
C
C   - calculate the total ic4-mole flows (kmol/h)
C
C   ISTATUS = FCA9809_ST * MW9809_ST * AI_9810_1_IC4_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     MFIC4 = ( FCA9809 / MW9809 ) * ( AI_9810_1_IC4
C   &      / 100 ) * 1000
C
C     MFIC4_ST = GOOD_STATUS
C   ELSE
C     MFIC4_ST = BAD_STATUS
C   ENDIF
C

```

```

C                                     ALKYTCALC.txt
C
C   ISTATUS = FCA9812_ST * MW9812_ST * AI_9820_1_IC4_ST
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     MFKIC4 = ( FCA9812 / MW9812 ) * ( AI_9820_1_IC4
C & / 100 ) * 1000
C
C     MFKIC4_ST = GOOD_STATUS
C   ELSE
C     MFKIC4_ST = BAD_STATUS
C   ENDIF
C
C
C   ISTATUS = FI9820_ST * MW9820_ST * ALKY_IC42_IC4_ST
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     MFKKIC4 = ( FI9820 / MW9820 ) * ( ALKY_IC42_IC4
C & / 100 ) * 1000
C
C     MFKKIC4_ST = GOOD_STATUS
C   ELSE
C     MFKKIC4_ST = BAD_STATUS
C   ENDIF
C
C
C   ISTATUS = FC9843_ST
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     MFKKKIC4 = ( FC9843 / MW9843 ) * MPIC43 * 1000
C   ELSE
C     MFKKKIC4 = 0.0
C   ENDIF
C
C   MFKKKIC4_ST = GOOD_STATUS
C
C .....
C
C - calculate the ic4-mole flows to the upper nozzles GD-9801A (kmo1/h)
C
C   ISTATUS = FC9873_ST * FCA9812_ST * FCA9809_ST
C & * MFIC4_ST * MFKIC4_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C     NUMER = FC9873 * ( MFIC4 + MFKIC4 )
C     DENOM = FCA9812 + FCA9809
C
C     MFIC4A = DIVCHK ( JPRNT, JDETAL,
C & NUMER, DENOM, IERROR )
C
C   ENDIF
C
C   IF ((ISTATUS .NE. BAD_STATUS) .AND.
C & (IERROR .EQ. GOOD_IER )) THEN
C
C     MFIC4A_ST = GOOD_STATUS
C   ELSE
C     MFIC4A_ST = BAD_STATUS
C   ENDIF
C
C
C   ISTATUS = FCA9881_ST * FI9820_ST
C & * MFKKIC4_ST * MFKKKIC4_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN

```

```

C                                     ALKYTCALC.txt
C      NUMER = FCA9881 * ( MFKKIC4 + MFKKKIC4 )
C      IF ( FC9843_ST .NE. BAD_STATUS ) THEN
C      DENOM = FI9820 + FC9843
C      ELSE
C      DENOM = FI9820
C      ENDIF
C      MFKKIC4A = DIVCHK ( JPRNT, JDETAL,
C &      NUMER, DENOM, IERROR )
C      ENDIF
C      IF ((ISTATUS .NE. BAD_STATUS) .AND.
C &      (IERROR .EQ. GOOD_IER )) THEN
C      MFKKIC4A_ST = GOOD_STATUS
C      ELSE
C      MFKKIC4A_ST = BAD_STATUS
C      ENDIF
C
C .....
C - calculate the ic4-mole flows to the lower nozzles GD-9801B (kmo1/h)
C      ISTATUS = MFIC4_ST * MFKIC4_ST * MFIC4A_ST
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C      MFIC4B = ( MFIC4 + MFKIC4 ) - MFIC4A
C      MFIC4B_ST = GOOD_STATUS
C      ELSE
C      MFIC4B_ST = BAD_STATUS
C      ENDIF
C
C      ISTATUS = MFKKIC4_ST * MFKKKIC4_ST * MFKKIC4A_ST
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C      MFKKIC4B = ( MFKKIC4 + MFKKKIC4 ) - MFKKIC4A
C      MFKKIC4B_ST = GOOD_STATUS
C      ELSE
C      MFKKIC4B_ST = BAD_STATUS
C      ENDIF
C      IF ( JPRINT .GT. 0 ) THEN
C      CALL PLABEL( ' ' )
C      CALL PLABEL('The ic4-mole flows')
C      CALL PLABEL('FIC-9809')
C      WRITE ( ILUNUM, 9040 ) 'MFIC4 (kmo1/h)' ,':', MFIC4
C      CALL PLABEL('FIC-9812')
C      WRITE ( ILUNUM, 9040 ) 'MFKIC4 (kmo1/h)' ,':', MFKIC4
C      CALL PLABEL('FI-9820')
C      WRITE ( ILUNUM, 9040 ) 'MFKKIC4 (kmo1/h)' ,':', MFKKIC4
C      CALL PLABEL('FIC-9843')
C      WRITE ( ILUNUM, 9040 ) 'MFKKKIC4 (kmo1/h)' ,':', MFKKKIC4
C      CALL PLABEL('The ic4 mole flows to the upper')
C      CALL PLABEL('nozzles GD-9801A')
C      WRITE ( ILUNUM, 9040 ) 'MFIC4A (kmo1/h)' ,':', MFIC4A
C      WRITE ( ILUNUM, 9040 ) 'MFKKIC4A (kmo1/h)' ,':', MFKKIC4A
C      CALL PLABEL('The ic4 mole flows to the lower')
C      CALL PLABEL('nozzles GD-9801B')
C      WRITE ( ILUNUM, 9040 ) 'MFIC4B (kmo1/h)' ,':', MFIC4B
C      WRITE ( ILUNUM, 9040 ) 'MFKKIC4B (kmo1/h)' ,':', MFKKIC4B
C      Page 32

```

```

                                ALKYTCALC.txt
      CALL PLABEL( ' ' )
C      ENDIF
C
C-----
C      I-B-4 The isobutane feed to the reactor, volume flows
C
C      IF ( JPRINT .GT. 0 ) THEN
C        CALL PLABEL( ' ' )
C        CALL PLABEL('I-B-4 The isobutane feed to the reactor,')
C        CALL PLABEL(' volume flows')
C        CALL PLABEL( ' ' )
C      ENDIF
C
C      - calculate the total ic4 volume flow to the reactor (m3/h)
C      ISTATUS = MFIC4_ST * MFKIC4_ST * MFKKIC4_ST * MFKKKIC4_ST
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        VAIC4 = ( MFIC4 + MFKIC4 + MFKKIC4 + MFKKKIC4 )
C        &      * MWIC4 / DNIC4
C
C        VAIC4_ST = GOOD_STATUS
C      ELSE
C        VAIC4_ST = BAD_STATUS
C      ENDIF
C
C-----
C      - calculate the ic4 volume flow to GD-9801A (m3/h)
C      ISTATUS = MFIC4A_ST * MFKKIC4A_ST
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        VAIC4A = ( MFIC4A + MFKKIC4A ) * MWIC4 / DNIC4
C
C        VAIC4A_ST = GOOD_STATUS
C      ELSE
C        VAIC4A_ST = BAD_STATUS
C      ENDIF
C
C-----
C      - calculate the ic4 volume flow to GD-9801B (m3/h)
C      ISTATUS = MFIC4B_ST * MFKKIC4B_ST
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        VAIC4B = ( MFIC4B + MFKKIC4B ) * MWIC4 / DNIC4
C
C        VAIC4B_ST = GOOD_STATUS
C      ELSE
C        VAIC4B_ST = BAD_STATUS
C      ENDIF
C
C      IF ( JPRINT .GT. 0 ) THEN
C
C        CALL PLABEL( ' ' )
C        CALL PLABEL('The total ic4-volume flow to the reactor')
C        WRITE ( ILUNUM, 9040 ) 'VAIC4 (m3/h)'      ', ', VAIC4
C        CALL PLABEL('The ic4-volumeflow to GD-9801A')
C        WRITE ( ILUNUM, 9040 ) 'VAIC4A (m3/h)'     ', ', VAIC4A
C        CALL PLABEL('The ic4-volumeflow to GD-9801B')
C        WRITE ( ILUNUM, 9040 ) 'VAIC4B (m3/h)'     ', ', VAIC4B
C        CALL PLABEL( ' ' )

```

```

C                                     ALKYTCALC.txt
C
C     ENDIF
C-----
C
C     I-C The isobutane to olefin (I/O) volume ratio
C
C     IF ( JPRINT .GT. 0 ) THEN
C       CALL PLABEL( ' ' )
C       CALL PLABEL( 'I-C The isobutane to olefin (I/O) volume ratio' )
C       CALL PLABEL( ' ' )
C     ENDIF
C
C     - the total isobutane to olefin (I/O) volume ratio from analysers (-)
C
C     ISTATUS = VAIC4_ST * VOLEF_ST
C
C     IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C       NUMER = VAIC4
C       DENOM = VOLEF
C
C       ALKYT_IC4OLE = DIVCHK ( JPRNT, JDETAL,
C & NUMER, DENOM, IERROR )
C     ENDIF
C
C     IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C & ( IERROR .EQ. GOOD_IER )) THEN
C
C       ALKYT_IC4OLE_ST = GOOD_STATUS
C     ELSE
C       ALKYT_IC4OLE_ST = BAD_STATUS
C     ENDIF
C-----
C
C     - the total isobutane to olefin (I/O) volume ratio from analysers
C     at the upper nozzles GD-9801A (-)
C
C     ISTATUS = VAIC4A_ST * VOLEFA_ST
C
C     IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C       NUMER = VAIC4A
C       DENOM = VOLEFA
C
C       ALKYT_IC4OLEA = DIVCHK ( JPRNT, JDETAL,
C & NUMER, DENOM, IERROR )
C     ENDIF
C
C     IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C & ( IERROR .EQ. GOOD_IER )) THEN
C
C       ALKYT_IC4OLEA_ST = GOOD_STATUS
C     ELSE
C       ALKYT_IC4OLEA_ST = BAD_STATUS
C     ENDIF
C-----
C
C     - the total isobutane to olefin (I/O) volume ratio from analysers
C     at the lower nozzles GD-9801B (-)
C
C     ISTATUS = VAIC4B_ST * VOLEFB_ST
C
C     IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C       NUMER = VAIC4B
C       DENOM = VOLEFB

```

```

C          ALKYTCALC.txt
C          ALKYT_IC4OLEB = DIVCHK ( JPRNT, JDETAL,
&          NUMER, DENOM, IERROR )
C          ENDIF
C          IF (( ISTATUS .NE. BAD_STATUS ) .AND.
&          ( IERROR .EQ. GOOD_IER )) THEN
C          ALKYT_IC4OLEB_ST = GOOD_STATUS
C          ELSE
C          ALKYT_IC4OLEB_ST = BAD_STATUS
C          ENDIF
C
C.....
C          - calculate the isobutane to olefin (I/O) ratio
C          for the upper nozzles GD-9801A (-)
C          ISTATUS = VOLEFA_ST * MFIC4A_ST * MFKKIC4_ST * MFKKKIC4_ST
&          * FI9820_ST
C          IF ( ISTATUS .NE. BAD_STATUS ) THEN
C          IF ( FC9843_ST .NE. BAD_STATUS ) THEN
C          NUMER = (( 7 * VOLEFA ) / MWIC4 * DNIC4 - MFIC4A )
&          * ( FI9820 + FC9843 )
C          ELSE
&          NUMER = (( 7 * VOLEFA ) / MWIC4 * DNIC4 - MFIC4A )
&          * ( FI9820 )
C          ENDIF
C          DENOM = MFKKIC4 + MFKKKIC4
C          IC4 = DIVCHK ( JPRNT, JDETAL,
&          NUMER, DENOM, IERROR )
C          ENDIF
C          IF (( ISTATUS .NE. BAD_STATUS ) .AND.
&          ( IERROR .EQ. GOOD_IER )) THEN
C          IC4_ST = GOOD_STATUS
C          ELSE
C          IC4_ST = BAD_STATUS
C          ENDIF
C
C          ISTATUS = FC9873_ST * FCA9809_ST * FCA9812_ST
C          IF ( ISTATUS .NE. BAD_STATUS ) THEN
C          NUMER = FC9873
C          DENOM = FCA9809 + FCA9812
C          K = DIVCHK ( JPRNT, JDETAL,
&          NUMER, DENOM, IERROR )
C          ENDIF
C          IF (( ISTATUS .NE. BAD_STATUS ) .AND.
&          ( IERROR .EQ. GOOD_IER )) THEN
C          K_ST = GOOD_STATUS
C          ELSE
C          K_ST = BAD_STATUS
C          ENDIF
C
C          ISTATUS = IC4_ST * K_ST * FCA9809_ST * FCA9812_ST

```

```

C                                     ALKYTCALC.txt
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C     NUMER = IC4 / 0.545 + K * FCA9809 / 0.525
C     DENOM = K * FCA9812 / 0.573
C     ALKYT_HC12 = DIVCHK ( JPRNT, JDETAL,
C   &     NUMER, DENOM, IERROR )
C   ENDIF
C
C   IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C   &     ( IERROR .EQ. GOOD_IER )) THEN
C     ALKYT_HC12_ST = GOOD_STATUS
C   ELSE
C     ALKYT_HC12_ST = BAD_STATUS
C   ENDIF
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL( ' ' )
C     CALL PLABEL('The calculation of ALKYT_HC12')
C     WRITE ( ILUNUM, 9040 ) 'IC4 (t/h)'          ',:', IC4
C     WRITE ( ILUNUM, 9040 ) 'K (-)'              ',:', K
C     CALL PLABEL( ' ' )
C   ENDIF
C
C .....
C
C   - calculate the isobutane to olefin (I/O) ratio
C   for the lower nozzles GD-9801B (-)
C
C   ISTATUS = MFKKIC4B_ST * VOLEF_ST * MFIC4_ST * MFKIC4_ST
C
C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
C     NUMER = MWIC4 / DNIC4 * MFKKIC4B
C     DENOM = ( 14 * VOLEF - ( MWIC4/DNIC4 ) * MFIC4
C   &     - (MWIC4/DNIC4) * MFKIC4 )
C     RATIO = DIVCHK ( JPRNT, JDETAL,
C   &     NUMER, DENOM, IERROR )
C   ENDIF
C
C   IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C   &     ( IERROR .EQ. GOOD_IER )) THEN
C     ALKYT_HC11 = 100 * ( 1 - RATIO)
C
C     ALKYT_HC11_ST = GOOD_STATUS
C   ELSE
C     ALKYT_HC11_ST = BAD_STATUS
C   ENDIF
C
C -----
C
C     I-D The additional isobutane feed needed
C
C   IF ( JPRINT .GT. 0 ) THEN
C     CALL PLABEL( ' ' )
C     CALL PLABEL('I-D The additional isobutane feed needed')
C     CALL PLABEL( ' ' )
C   ENDIF
C
C   ISTATUS = VOLEF_ST * MFIC4_ST
C
C
C

```

```

                                ALKYTCALC.txt
IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C - calculate the olefin mole flow (kmol/h)
C
MFOLEF = VOLEF * DNC4T / MWC4T
C
C - calculate the needed fresh feed of isobutane according to the
C fresh feed of olefin (t/h). The ratio is 1 isobutane : 1 olefin.
C
ALKYT_LISAIC4 = ( MFOLEF - MFIC4 - MFKIC4 )
& * MWIC4 / DNIC4
C
ALKYT_LISAIC4_ST = GOOD_STATUS
ELSE
ALKYT_LISAIC4_ST = BAD_STATUS
ENDIF
C
IF ( JPRINT .GT. 0 ) THEN
CALL PLABEL( ' ' )
CALL PLABEL('The olefin mole flow')
WRITE (ILUNUM,9040) 'MFOLEF (kmol/h)' ,':', MFOLEF
ENDIF
C
-----
C
C II YIELDS AND MATERIAL BALANCE CLOSURE
C II-A The complete Alkylation unit
C
IF ( JPRINT .GT. 0 ) THEN
CALL PLABEL( ' ' )
CALL PLABEL('II Yields and material balance')
CALL PLABEL(' closure')
CALL PLABEL('II-A The complete Alkylation unit')
CALL PLABEL( ' ' )
ENDIF
C
C - calculate the stream yields and the material balance closure for the
C complete Alkylation unit. Calculate the total feed flow from component
C flows then the yield from product flows. The mass balance is checked
C for the closure by summing the yields.
C
ISTATUS = FCA9809_ST * FCA9812_ST * FC9852_ST
& * FC9836_ST * FC9849_ST * FC9845_ST
C
IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
FLOW(1) = FCA9809
FLOW(2) = FCA9812
FLOW(3) = FC9852
FLOW(4) = FC9836
FLOW(5) = FC9849 - FC9845
C
SID(1) = 1
SID(2) = 1
SID(3) = -1
SID(4) = -1
SID(5) = -1
C
CALL BALYD ( JPRINT, JDETAL, S, FLOW, SID, Y,
& ALKYT_MB_CL, IERROR )
C
ENDIF
C
IF (( ISTATUS .NE. BAD_STATUS ) .AND.
& ( IERROR .EQ. GOOD_IER )) THEN
C
ALKYT_IC4F_MY = Y(1)
ALKYT_OLEFF_MY = Y(2)

```

```

                                ALKYTCALC.txt
    ALKYT_NC4_MY   = Y(3)
    ALKYT_ALKY_MY  = Y(4)
    ALKYT_PRA_MY   = Y(5)
C
    ALKYT_IC4F_MY_ST = GOOD_STATUS
    ALKYT_OLEFF_MY_ST = GOOD_STATUS
    ALKYT_NC4_MY_ST  = GOOD_STATUS
    ALKYT_ALKY_MY_ST  = GOOD_STATUS
    ALKYT_PRA_MY_ST  = GOOD_STATUS
    ALKYT_MB_CL_ST   = GOOD_STATUS
ELSE
    ALKYT_IC4F_MY_ST = BAD_STATUS
    ALKYT_OLEFF_MY_ST = BAD_STATUS
    ALKYT_NC4_MY_ST  = BAD_STATUS
    ALKYT_ALKY_MY_ST  = BAD_STATUS
    ALKYT_PRA_MY_ST  = BAD_STATUS
    ALKYT_MB_CL_ST   = BAD_STATUS
ENDIF
C
C-----
C
C      II-B DA-9801
C
C      IF ( JPRINT .GT. 0 ) THEN
C        CALL FLABEL( ' ' )
C        CALL FLABEL( 'II-B DA-9801' )
C        CALL FLABEL( ' ' )
C      ENDIF
C
C      - calculate the stream yields and the material balance closure for the
C      DA-9801A unit. Calculate the total feed flow from component
C      flows then the yield from product flows. The mass balance is checked
C      for the closure by summing the yields.
C
C      ISTATUS = FC9801_ST * FC9805_ST * FC9806_ST
C      &        = FCA9809_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        FLOW(1) = FC9801
C        FLOW(2) = FC9806
C        FLOW(3) = FC9805
C        FLOW(4) = FCA9809
C
C        SID(1) = 1
C        SID(2) = -1
C        SID(3) = -1
C        SID(4) = -1
C
C        CALL BALYD ( JPRNT, JDETAL, 4, FLOW, SID, Y,
C      &              ALKYT_D01MB_CL, IERROR )
C
C      ENDIF
C
C      IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C      &    ( IERROR .EQ. GOOD_IER )) THEN
C
C        ALKYT_D01F_MY   = Y(1)
C        ALKYT_D01_NC4_MY = Y(2)
C        ALKYT_D01_IC4VAR_MY = Y(3)
C        ALKYT_D01_IC4_MY  = Y(4)
C
C        ALKYT_D01F_MY_ST   = GOOD_STATUS
C        ALKYT_D01_NC4_MY_ST = GOOD_STATUS
C        ALKYT_D01_IC4VAR_MY_ST = GOOD_STATUS
C        ALKYT_D01_IC4_MY_ST  = GOOD_STATUS
C        ALKYT_D01MB_CL_ST  = GOOD_STATUS
ELSE

```

```

                                ALKYTCALC.txt
ALKYT_D01F_MY_ST      = BAD_STATUS
ALKYT_D01_NC4_MY_ST  = BAD_STATUS
ALKYT_D01_IC4VAR_MY_ST = BAD_STATUS
ALKYT_D01_IC4_MY_ST  = BAD_STATUS
ALKYT_D01MB_CL_ST    = BAD_STATUS
ENDIF
C
C-----
C
C      III  PRODUCT COMPOSITION CALCULATIONS
C      III-A The butane splitter DA-9801
C
C      IF ( JPRINT .GT. 0 ) THEN
C        CALL PLABEL( ' ' )
C        CALL PLABEL( 'III  Product composition calculations' )
C        CALL PLABEL( 'III-A The butane splitter DA-9801' )
C      ENDIF
C
C      - calculate the amount of isobutane which is going to the storage.
C      when the valve is closed then zeroing the value.
C
C      ISTATUS = FC9805_ST * FC9805_VE_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        IF ( FC9805_VE .GT. 1 ) THEN
C          ALKYT_IC4VAR = FC9805
C        ELSE
C          ALKYT_IC4VAR = 0
C        ENDIF
C
C        ALKYT_IC4VAR_ST = GOOD_STATUS
C      ELSE
C        ALKYT_IC4VAR_ST = BAD_STATUS
C      ENDIF
C
C-----
C
C      - calculate the total amount of the products
C
C      ISTATUS = FC9806_ST * ALKYT_IC4VAR_ST * FCA9809_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        ALKYT_TU0TC4 = FC9806 + ALKYT_IC4VAR + FCA9809
C
C      - calculate the percentage values of products from DA-9801
C
C      IF ( ALKYT_TU0TC4 .GT. 1 ) THEN
C        ALKYT_F980605 = FC9806 / ALKYT_TU0TC4 * 100
C        ALKYT_IC4VAR05 = ALKYT_IC4VAR / ALKYT_TU0TC4 * 100
C        ALKYT_F980905 = FCA9809 / ALKYT_TU0TC4 * 100
C      ELSE
C        ALKYT_F980605 = 0
C        ALKYT_IC4VAR05 = 0
C        ALKYT_F980905 = 0
C      ENDIF
C
C      - calculate the sum of the percentage values of products
C
C      ALKYT_TU0TC405 = ALKYT_F980605 + ALKYT_IC4VAR05 + ALKYT_F980905
C
C      ALKYT_TU0TC4_ST = GOOD_STATUS
C      ALKYT_IC4VAR05_ST = GOOD_STATUS
C      ALKYT_F980605_ST = GOOD_STATUS
C      ALKYT_F980905_ST = GOOD_STATUS
C      ALKYT_TU0TC405_ST = GOOD_STATUS
C    ELSE
C      ALKYT_TU0TC4_ST = BAD_STATUS

```

```

                                ALKYTCALC.txt
    ALKYT_IC4VAROS_ST = BAD_STATUS
    ALKYT_F980605_ST = BAD_STATUS
    ALKYT_F980905_ST = BAD_STATUS
    ALKYT_TU0TC405_ST = BAD_STATUS
ENDIF
C
C-----
C
C      III-B The main fractionator DA-9802
C
C      IF ( JPRINT .GT. 0 ) THEN
C        CALL PLABEL( ' ' )
C        CALL PLABEL( 'III-B The main fractionator DA-9802' )
C        CALL PLABEL( ' ' )
C      ENDIF
C
C      - calculate the amount of butane which is going to the storage.
C      When the temperature in the butane sidedraw is under 80 degree
C      Celsius then zeroing the value.
C
C      ISTATUS = FC9852_ST * TI9865_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        IF ( TI9865 .GT. 80 ) THEN
C          ALKYT_NC4VAR = FC9852
C        ELSE
C          ALKYT_NC4VAR = 0
C        ENDIF
C
C        ALKYT_NC4VAR_ST = GOOD_STATUS
C      ELSE
C        ALKYT_NC4VAR_ST = BAD_STATUS
C      ENDIF
C
C-----
C
C      - calculate the amount of propane to the storage
C
C      ISTATUS = FC9849_ST * FC9845_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        IF ( FC9849 .GT. FC9845 ) THEN
C          ALKYT_C3VAR = FC9849 - FC9845
C        ELSE
C          ALKYT_C3VAR = 0
C        ENDIF
C
C        ALKYT_C3VAR_ST = GOOD_STATUS
C      ELSE
C        ALKYT_C3VAR_ST = BAD_STATUS
C      ENDIF
C
C-----
C
C      - calculate the total amount of the products. The volume flow of
C      alkylate which is going to the storage has to be over 5 t/h.
C
C      ISTATUS = FC9836_ST * ALKYT_C3VAR_ST * ALKYT_NC4VAR_ST
C
C      IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C        IF ( FC9836 .GT. 5 ) THEN
C          ALKYT_TU0TALKY = ALKYT_C3VAR + ALKYT_NC4VAR + FC9836
C        ELSE
C          ALKYT_TU0TALKY = 0
C        ENDIF

```

```

C                                     ALKYTCALC.txt
C
C - calculate the percentage values of products from DA-9802
C
C     IF ( ALKYT_TUOTALKY .GT. 1 ) THEN
C       ALKYT_C3VAROS = ALKYT_C3VAR / ALKYT_TUOTALKY * 100
C       ALKYT_NC4VAROS = ALKYT_NC4VAR / ALKYT_TUOTALKY * 100
C       ALKYT_F98360S = FC9836 / ALKYT_TUOTALKY * 100
C     ELSE
C       ALKYT_C3VAROS = 0
C       ALKYT_NC4VAROS = 0
C       ALKYT_F98360S = 0
C     ENDIF
C
C - calculate the sum of the percentage values of products
C
C     ALKYT_TUOTALKYOS = ALKYT_C3VAROS + ALKYT_NC4VAROS
C   &
C     + ALKYT_F98360S
C
C     ALKYT_TUOTALKY_ST = GOOD_STATUS
C     ALKYT_C3VAROS_ST = GOOD_STATUS
C     ALKYT_NC4VAROS_ST = GOOD_STATUS
C     ALKYT_F98360S_ST = GOOD_STATUS
C     ALKYT_TUOTALKYOS_ST = GOOD_STATUS
C   ELSE
C     ALKYT_TUOTALKY_ST = BAD_STATUS
C     ALKYT_C3VAROS_ST = BAD_STATUS
C     ALKYT_NC4VAROS_ST = BAD_STATUS
C     ALKYT_F98360S_ST = BAD_STATUS
C     ALKYT_TUOTALKYOS_ST = BAD_STATUS
C   ENDIF
C
C-----
C
C     III-C Reflux/distillate-ratios
C     III-C-1 The butane splitter DA-9801
C
C     IF ( JPRINT .GT. 0 ) THEN
C       CALL PLABEL( ' ' )
C       CALL PLABEL( 'III-C Reflux/distillate-ratios' )
C       CALL PLABEL( 'III-C-1 The butane splitter DA-9801' )
C       CALL PLABEL( ' ' )
C     ENDIF
C
C     ISTATUS = FCA9804_ST * FC9805_ST * FCA9809_ST
C
C     IF ( ISTATUS .NE. BAD_STATUS ) THEN
C
C       NUMER = FCA9804
C       DENOM = FC9805 + FCA9809
C
C       ALKYT_D01RF2DIST = DIVCHK ( JPRNT, JDETAL,
C   &
C     NUMER, DENOM, IERROR )
C
C     ENDIF
C
C     IF (( ISTATUS .NE. BAD_STATUS ) .AND.
C   &
C     ( IERROR .EQ. GOOD_IER )) THEN
C
C       ALKYT_D01RF2DIST_ST = GOOD_STATUS
C     ELSE
C       ALKYT_D01RF2DIST_ST = BAD_STATUS
C     ENDIF
C
C-----
C
C     III-C-2 The main fractionator DA-9802
C
C     IF ( JPRINT .GT. 0 ) THEN
C       CALL PLABEL( ' ' )

```

```

                                ALKYTCALC.txt
      CALL PLABEL('III-C-2 The main fractionator DA-9802')
      CALL PLABEL('')
    C   ENDIF
    C   ISTATUS = FCSA9837_ST * FCSA9838_ST
    C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
    C
    C     NUMER = FCSA9837
    C     DENOM = FCSA9838
    C
    C     ALKYT_D02RF2DIST = DIVCHK ( JPRNT, JDETAL,
    C   &     NUMER, DENOM, IERROR )
    C   ENDIF
    C   IF (( ISTATUS .NE. BAD_STATUS ) .AND.
    C   &     ( IERROR .EQ. GOOD_IER )) THEN
    C
    C     ALKYT_D02RF2DIST_ST = GOOD_STATUS
    C   ELSE
    C     ALKYT_D02RF2DIST_ST = BAD_STATUS
    C   ENDIF
    C
    C-----
    C   IV   PCT CALCULATIONS
    C
    C   IF ( JPRINT .GT. 0 ) THEN
    C     CALL PLABEL('')
    C     CALL PLABEL
    C   &     ('IV   PCT Calculations')
    C     CALL PLABEL('')
    C   ENDIF
    C
    C   Calculate PCT at bottom 9 using empirical linear correlation
    C
    C   ISTATUS = TI9808_ST * PCA9802_ST
    C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
    C
    C     ALKYT_D01PCT09
    C   &     = TI9808 - ( PCA9802 - 550 ) * 0.0543
    C
    C     ALKYT_D01PCT09_ST = GOOD_STATUS
    C   ELSE
    C     ALKYT_D01PCT09_ST = BAD_STATUS
    C   ENDIF
    C
    C-----
    C   V   MISCELLANEOUS CALCULATIONS
    C
    C   IF ( JPRINT .GT. 0 ) THEN
    C     CALL PLABEL('')
    C     CALL PLABEL
    C   &     ('V   Miscellaneous Calculations')
    C     CALL PLABEL('')
    C   ENDIF
    C
    C   Calculate ALKYT_PDICVE
    C
    C   ISTATUS = PDC9804_VE_ST
    C   IF ( ISTATUS .NE. BAD_STATUS ) THEN
    C

```



```

C                                     ALKYTCALC.txt
C
C   CONSTRAINT LIMIT COORDINATION COMES IN HERE - (IF ANY)
C-----
C
C   - Constraint Limit Coordination
C
C   This section will be commented out if there is no constraint
C   coordination done in this program.
C
C   IF (JPRINT .GT. 0) THEN
C     CALL PLABEL ( ' ' )
C     CALL PLABEL
C   & ( ' Table: xx-xx MV Constraint Limit Coordination ' )
C     CALL PLABEL ( ' ' )
C   ENDIF
C
C   BJH_PC33 = BJH_PC31
C   BJH_PC32 = BJH_PC31
C-----
C=====
C-----
C
C   IF (JPRINT .GT. 0) THEN
C     CALL PLABEL ( ' ' )
C     CALL PLABEL
C   & ( ' Check the watchdog ' )
C     CALL PLABEL ( ' ' )
C   ENDIF
C-----
C
C   - Set the DEBUG Print Flag to Zero
C   - Set the Valid Calculation Flag to Five
C
C   IF ( ALKYT_CALC_PR .NE. 0 ) THEN
C     ALKYT_CALC_PR = ALKYT_CALC_PR - 1
C   ENDIF
C
C   ALKYT_VALIDCALC = 5
C-----
C
C   - watchdog for G2 connection is "multiplied" here for the
C   separate CMI controllers.
C
C   ALKYG_WDOG = ALKYG_WDOG - 1
C   IF (ALKYG_WDOG .LT. 1) ALKYG_WDOG = 0
C-----
C
C   - Watchdogs for CMI controllers
C
C   IF (ALKYG_WDOG .GT. 0) THEN
C     ALKYCMI_F9802_WDOG = 5
C     ALKYCMI_F9802_WDOG_ST = GOOD_STATUS
C
C     ALKYCMI_F9804_WDOG = 5
C     ALKYCMI_F9804_WDOG_ST = GOOD_STATUS
C
C     ALKYCMI_F9803_WDOG = 5
C     ALKYCMI_F9803_WDOG_ST = GOOD_STATUS
C
C   ELSE

```



```

CC          ALKYTCALC.txt
CC          IFERR = 340
CC
CC          DO I = 1, IDOMX
CC              IF ( IDOST(I) .EQ. BAD_STATUS ) THEN
CC                  INT_VDO(I) = 0          ! Bad status
CC              ELSEIF ( INT_VDO(I) .EQ. FALSE_VALUE ) THEN
CC                  INT_VDO(I) = 1          ! False and good
CC              ELSE
CC                  INT_VDO(I) = 2          ! True and good
CC              ENDIF
CC          ENDDO
CC          CALL DBSETD      (IDOMX      ,      ! Number of variables
CC          &                IDO        ,      ! Indices database variables
CC          &                INT_VDO    ,      ! Value of digital output
CC          &                IERR       )      ! Error return code
CC
CC          IF ( IERR .NE. 0 ) THEN
CC              CALL PLABEL
CC          &          ( ' Error writing digital values to database ' )
CC              GOTO 8100
CC          ENDIF
CC
CC          .....
CC
CC          - write the print and level flags to the database
CC          IF ( JPRINT .GT. 0 ) THEN
CC              CALL PLABEL( ' ' )
CC              CALL PLABEL( ' Write print and level flags' )
CC          ENDIF
CC
CC          IFERR = 350
CC          CALL DBSETI ( IIMEMX      ,      ! number of variables
CC          &                IIME      ,      ! table for Tagnames of variables
CC          &                IIME      ,      ! table for values of variables
CC          &                IERR       )      ! error return code
CC          IF ( IERR .NE. 0 ) THEN
CC              CALL PLABEL
CC          &          ( ' Error writing print and level flags to database ' )
CC              GO TO 8100
CC          ENDIF
CC
CC          .....
CC
CC          Set Filtering Flag To 1
CC          INIEXI = 1
CC
CC          .....
CC          C$DEBUG PRINT
CC
CC          .....
CC          - Print the Results to the DEBUG PRINT File
CC
CC          .....
CC          - Print the Database Outputs
CC          IF ( JPRINT .GT. 0 ) THEN
CC              CALL PLABEL ( ' ' )

```

```

                                ALKYTCALC.txt
CALL LABEL ( ' PROGRAM OUTPUTS ' )
CALL LABEL ( ' ' )
C
CALL LABEL ( ' ' )
CALL LABEL ( '***** CALCULATED VALUES' )
IFERR = 470
CALL DBTEXT ( ICAMX, ICA, %REF(CATEXT), IERR )
IF ( IERR .NE. 0 ) THEN
CALL MESSAGES
&      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
ENDIF
DO I = 1, ICAMX
WRITE(ILUNUM, 9000) CATEXT(I)( 1:20), CATEXT(I)(21:52),
&      CA(I), CATEXT(I)(53:60), ICAST(I)
ENDDO
C
CALL LABEL ( ' ' )
CALL LABEL ( '***** NEW FILTERED VALUES AND FILTER FACTORS' )
IFERR = 480
CALL DBTEXT ( IFVMX, IFV, %REF(FVTEXT), IERR )
IF ( IERR .NE. 0 ) THEN
CALL MESSAGES
&      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
ENDIF
DO I = 1, IFVMX
WRITE(ILUNUM, 9001) FVTEXT(I)( 1:20), FVTEXT(I)(21:52),
&      FV(I), 'FACTOR = ', FF(I)
ENDDO
C
CALL LABEL ( ' ' )
CALL LABEL ( '***** INTEGER MANUAL ENTRIES' )
IFERR = 490
CALL DBTEXT ( IIMEX, IIME, %REF(IMETEXT), IERR )
IF ( IERR .NE. 0 ) THEN
CALL MESSAGES
&      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
ENDIF
DO I = 1, IIMEX
WRITE(ILUNUM, 9000) IMETEXT(I)( 1:20), IMETEXT(I)(21:52),
&      FLOAT(IIME(I)), IMETEXT(I)(53:60), 1
ENDDO
C
CALL LABEL ( ' ' )
CALL LABEL ( '***** DIGITAL OUTPUTS ' )
CALL LABEL ( '0=BAD, 1=GOOD AND FALSE, 2=GOOD AND TRUE ' )
IFERR = 500
CALL DBTEXT ( IDOMX, IDO, %REF(DOTEXT), IERR )
IF ( IERR .NE. 0 ) THEN
CALL MESSAGES
&      (IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)
ENDIF
DO I = 1, IDOMX
WRITE(ILUNUM, 9000) DOTEXT(I)( 1:20), DOTEXT(I)(21:52),
&      VDO(I), DOTEXT(I)(53:60), IDOST(I)
ENDDO
C
C-----
C
END IF
GOTO 9991
C
*****
C
*** ERROR HANDLING ***
*****
C
8000 CONTINUE
8100 CONTINUE
CALL MESSAGES(IURGALARM, IDBPG, IDBPA, 0, IERR, IFERR, 730, MERR)

```

```
                                ALKYTCALC.txt
GOTO 9991

C-----
C
C - WRITE STATEMENT FORMATS
C
9000 FORMAT(1X, A20, 2X, A32, 3X, G12.6, 2X, A8, 3X, I1)
9001 FORMAT(1X, A20, 2X, A32, 3X, G12.6, 2X, A16, G12.6)
9002 FORMAT(1X, A20, 2X, A32, 3X, G12.6, 2X, I1)
9003 FORMAT(1X, 'PROGRAM: ALKYTCALC', 3X, 'DATE: ', A9, 3X, 'TIME: ', A8)
9005 FORMAT(1X, A20, 2X, A32, 3X, G12.6, 2X, A16, G12.6, 1X, I1)
C
9040 FORMAT(1X, A30, 1X, A2, 1X, G12.6)
C-----
C
C - Close the DEBUG PRINT File
C
C.....
C
9991 CONTINUE
C
IF ( JPRINT .GT. 0 ) THEN
  CALL DATE_AND_TIME( CDATE, CTIME )
C
  CALL PLABEL ( : : )
  CALL PLABEL ( : : )
  CALL PLABEL('+++++')
  WRITE(ILUNUM, 9003) CDATE, CTIME
  CALL PLABEL('          END OF ALKY CALCULATIONS ')
  CALL PLABEL('+++++')
C
  CLOSE ( ILUNUM )
  JPRINT = 0
END IF
C-----
C
C - hibernate; wake-up is scheduled at regular intervals in run command
C
C.....
C
9995 CONTINUE
C
  ICOND = SYS$HIBER()
  GO TO 1
C-----
C
C                                EXIT PROGRAM
C-----
C
C                                END
C-----
```

