

---

**5. DESENVOLUPAMENT DEL PROGRAMARI  
DE CONTROL DEL PROCÉS.**

---

## 5. DESENVOLUPAMENT DEL PROGRAMARI DE CONTROL DEL PROCÉS.

### 5.1. MOTIVACIONS DE LA SEVA CONSTRUCCIÓ.

Com s'ha destacat al capítol d'*Introducció*, és necessari disposar de mesures en línia del comportament fisiològic i metabòlic cel·lular per poder plantejar-se possibles estratègies de control del procés. De forma convencional, els bioreactors incorporen sistemes de mesura i control de les variables bàsiques, com la temperatura, el pH, l'oxigen dissolt, l'agitació i, en alguns casos, el potencial redox. D'aquesta manera, es pot assegurar que els cultius es porten a terme en les condicions ambientals requerides, però quan es volen desenvolupar estratègies de control més complexes, en què l'acció de control es basa en una mesura no convencional, com en aquest treball ho són la concentració i l'activitat cel·lular, i implica la manipulació d'un o varis corrents d'aliment per a l'addició controlada de nutrients (cas de l'estratègia de cultiu en *fed-batch*), aleshores esdevé indispensable un sistema de control de totes les variables, centralitzades en un ordinador de gestió (Kong i col., 1998). Per aquesta raó, és necessari construir un sistema de control on es puguin integrar les mesures de tot tipus que es fan al bioreactor (tant les de variables de l'entorn com les implementades en el treball) i es puguin definir i executar estratègies de control, per tant, que permetin l'actuació sobre una sèrie d'equips addicionals (bombes, electrovàlvules, etc.). A la bibliografia es troben exemples de sistemes de control que s'han aplicat en cultius cel·lulars (Büntemeyer i col., 1994; Kyung i col., 1994; Konstantinov i col., 1996). L'àmplia disponibilitat de targetes d'adquisició de dades i d'eines o llenguatges de programació orientats a objectes, amb combinació d'alguns components independents, com subsistemes, controladors d'aeració, bombes, etc., fa molt més assequible el disseny d'aplicacions per controlar el procés.

### 5.2. DEFINICIÓ DELS REQUERIMENTS.

Per al desenvolupament del programari de control del procés, serà necessari prendre mesures en línia d'una sèrie de variables del procés, així com gestionar tota una sèrie d'equips i elements associats al cultiu amb l'objectiu final de controlar el procés. En aquest treball, algunes de les mesures es realitzen directament a través dels sensors i la unitat de control digital (DCU) de què disposa l'equip de cultiu, com per exemple el pH, mentre que altres variables, s'obtenen mitjançant una seqüència d'instruccions sobre el bioreactor i l'equip associat al mateix, com és el cas de la mesura de la OUR. La durada del procés pot anar des d'uns dies, per a un experiment en discontinu, a varies setmanes, en el cas d'un procés en

continu. Durant aquest temps, s'han de fer periòdicament anàlisis automàtics, s'han de processar dades, s'han de guardar les lectures i els resultats, i s'han de realitzar les accions de control en funció de les dades processades i la llei o algoritme de control corresponent. També cal tenir, per a cada tipus d'operació, un control remot total sobre el muntatge experimental. Finalment, cal disposar d'una interfície d'usuari que faciliti la lectura i la visualització dels paràmetres del procés, tant els adquirits com els processats, i doni a l'usuari la possibilitat de poder interaccionar en qualsevol moment sobre el programa de control. Per tant, l'objectiu principal és construir una eina robusta, fiable i fàcilment modificable (aplicació oberta), per poder configurar, comprovar i validar les diferents estratègies d'operació, i les tècniques d'instrumentació i control del cultiu amb cèl·lules animals.

### 5.3. CRITERIS DE SELECCIÓ DE L'ENTORN DE PROGRAMACIÓ.

Durant la fase d'anàlisi de les especificacions, es van proposar, quant a l'entorn de desenvolupament, tres alternatives per a la confecció del programari de control del procés:

- Visual C++: És un entorn de treball basat en la programació orientada a objectes amb el llenguatge C++. La programació amb aquest entorn és molt complexa, i no la fa apropiada a la necessitat de què pugui ser fàcilment modificable i ampliable per l'usuari. Un altre dels problemes que suposa la utilització del l'entorn Visual C++ és l'elevat temps que s'ha d'invertir en el desenvolupament d'interfícies d'usuari.
- LabWindows/CVI: És un producte de l'empresa National Instruments, que es dedica principalment a la fabricació d'equips i programari per a la mesura i l'anàlisi des d'ordinadors. LabWindows/CVI és un entorn de desenvolupament que proporciona una sèrie de biblioteques amb funcions d'alt nivell, així com eines i controls específics per desenvolupar aplicacions orientades al treball amb instrumentació. El llenguatge que utilitza és l'estàndard ANSI C (a l'apartat 8.4.2 es descriuen més característiques sobre aquest entorn, i també es pot accedir directament als manuals, consultant la pàgina Web: <http://www.natinst.com>).
- LabVIEW: És un altre dels productes de l'empresa National Instruments, orientat al desenvolupament d'aplicacions per a l'adquisició de dades i el control. Disposa de les mateixes eines gràfiques i biblioteques que el LabWindows/CVI. La gran diferència d'aquest entorn és que desenvolupa les aplicacions amb llenguatge de programació gràfica. La programació gràfica té l'avantatge que és fàcil i intuïtiva, i es fonamenta en la interconnexió de blocs (o instruments), essent una clara analogia del mode de treball en electricitat, electrònica, etc. El desavantatge principal és que les aplicacions realitzades amb aquest entorn són complicades de mantenir sobretot si es tracta d'aplicacions grans.

Després de valorar els avantatges i desavantatges de cadascun dels entorns de programació per al compliment dels objectius marcats, i tenint en compte la familiaritat amb el llenguatge C, es va decidir que LabWindows/CVI era l'eina més indicada, en aquest cas, per desenvolupar les aplicacions *Sequen* i *Supervis*. Aquestes aplicacions han d'incloure els següents elements: interfície d'usuari, control lògic del programa, adquisició de dades i anàlisi de dades (*Sequen*), visualització de les dades rellevants del procés (*Supervis*).

#### 5.4. ESPECIFICACIONS TÈCNIQUES.

Per assolir el requisits comentats a l'apartat 5.2, l'ordinador de control s'ha de connectar a la unitat de control digital (DCU) del Biostat MCD de Braun que conté les bombes dosificadora FE-211 i peristàltica FOAM. A més a més, s'ha de connectar a 4 electrovàlvules de 3-vies de Sirai, a 4 bombes peristàltiques d'Ismatec i al turbidímetre AS82 d'Aquasant. A la Figura 5.1 es mostra, a mode d'exemple, un esquema d'un dels muntatges experimentals realitzat en aquest treball, en què es poden observar diferents comunicacions entre l'ordinador i l'equip Biostat MCD o la diferent instrumentació associada al cultiu.

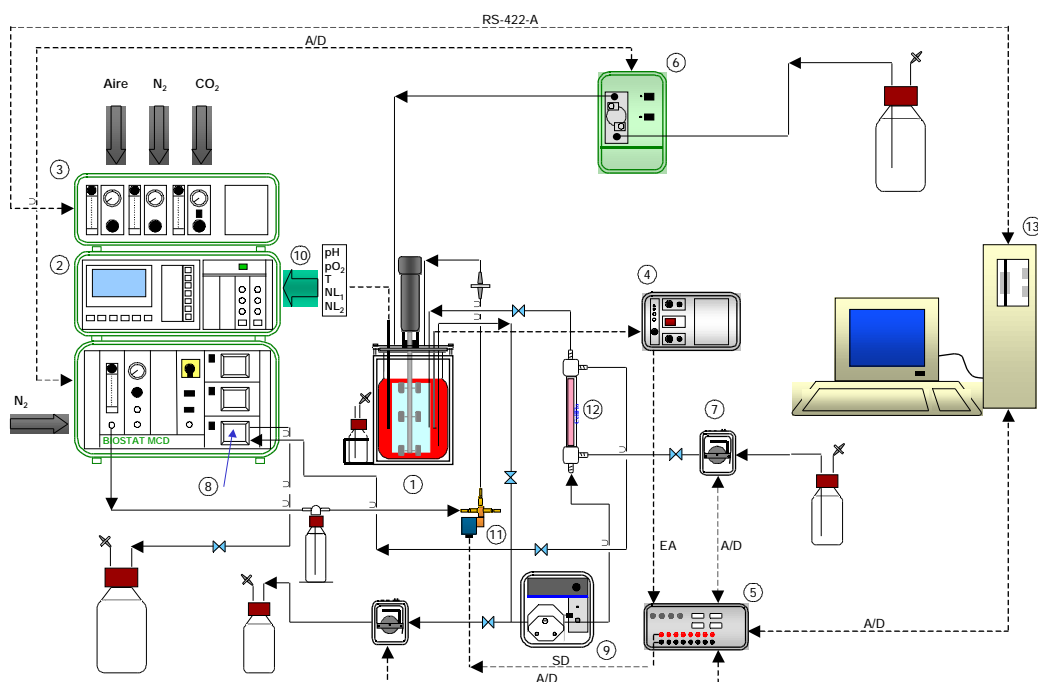


Figura 5.1. Esquema de l'adquisició de dades i control de l'operació en perfusió. 1) Bioreactor Braun Biostat MCD, 2) Unitat de control digital (DCU), 3) Estació de mescla de gasos  $pO_2$  Gas-Flow, 4) Sonda Aquasant AF44 i transmissor AS82, 5) Caixa de comunicacions, 6) Bomba Braun FE-211, 7) Bomba Ismatec, 8) Bomba FOAM, 9) Bomba Masterflex L/S, 10) Sondes de temperatura, pH,  $pO_2$  i nivell ( $NL_1$  i  $NL_2$ ), 11) Electrovàlvula Sirai, 12) Mòdul de microfiltració Microgon CellFlo, 13) Ordinador de gestió i control. A/D = Sortides i Entrades Analògiques/Digitals, EA = Entrada Analògica, SD = Sortida Digital, RS-422-A = port sèrie.

En un altre treball efectuat dins del mateix grup de recerca, també s'ha desenvolupat aquest programari per poder incorporar mesures en línia de components clau del medi (glucosa, lactat, glutamina i amoni), mesurats a través d'un equip d'anàlisi FIA comercial (ASIA), o bé, amb un equip de vàlvules per mòduls (TMI), i detectats espectrofotomètrica i amperomètricament. A la Taula 5.1 es presenten un resum de totes les comunicacions que s'estableixen entre l'ordinador i els diferents equips, així com les funcions en què s'han destinat cadascun d'aquests equips, tot i que en aquest treball aquelles que corresponen als equips FIA no es tracten.

Entitat	Descripció	Comunicacions	Unitats
Bombes Ismatec	Bomba peristàtica de velocitat variable i reversible, utilitzada per bombejar diferents solucions en els cultius de cèl·lules animals.	Analògic i digital	4
DCU	Unitat de Control Digital que es connecta a la targeta RS-422-A Opto22 AC24AT, que controla totes les variables bàsiques de l'entorn del bioreactor.	RS-422-A	2
Turbidímetre	Sensor Aquasant AF44 CS/R amb el transmissor AS82, que mesura la densitat cel·lular del cultiu cel·lular.	Analògic	1
Espectrofotòmetre	Sensor espectrofotomètric que pot connectar-se a l'ASIA o la targeta d'adquisició de dades Advantech PCL-812-PG, que permet fer mesures dels substrats amoni i glutamina.	Analògic o RS-232-C	1
Amperòmetre	Sensor amperomètric que pot connectar-se a l'ASIA o la targeta d'adquisició de dades Advantech PCL-812-PG, que permet fer mesures dels substrats lactat i glucosa.	Analògic o RS-232-C	1
Electrovàlvula	Vàlvules de 3-vies accionades per senyal digital (TTL) des de l'ordinador, que s'utilitzen per obrir o tancar el pas de línies de líquid o gas.	Digital	4
Bomba FOAM	Bomba peristàtica de velocitat fixa, que s'utilitza per efectuar el rentat del sensor Aquasant, o bé per controlar el nivell de líquid en el bioreactor.	RS-422-A	1
Bomba FE-211	Bomba dosificadora de membrana de velocitat variable, que es fa servir per aportar la quantitat necessària de nutrients en les diferents estratègies que ho requereixen.	RS-422-A o Analògic i Digital	1
TMI-6WAYV	Vàlvula de 6 vies, que serveix per repartir el flux en la seqüència d'anàlisi de substrats.	RS-485	4
TMI-INJV	Vàlvula d'Injecció, amb dues posicions, de càrrega o d'injecció, que serveix per inserir la mostra en un altre circuit d'injecció.	RS-485	2
FIA-ASIA	Analitzador d'injecció en flux ASIA, que permet fer l'anàlisi de substrats i al que es poden connectar diferents detectors.	RS-232-C	1

Taula 5.1. Breu descripció dels diferents elements a comunicar amb l'ordinador de gestió i control.

## 5.5. FUNCIONS I DISSENY TÈCNIC.

El programa Sequen.prj és una aplicació desenvolupada en l'entorn de LabWindows/CVI que permet gestionar i processar tota la informació de la diferent instrumentació, així com actuar sobre el procés. Bàsicament, el funcionament d'aquesta aplicació es pot simplificar en tres blocs:

1. Actuació remota i control manual sobre els equips associats al procés.
2. Adquisició de dades a intervals predeterminats de temps programables.
3. Execució, des d'un arxiu de text, d'una seqüència que conté totes les ordres a realitzar del procés.

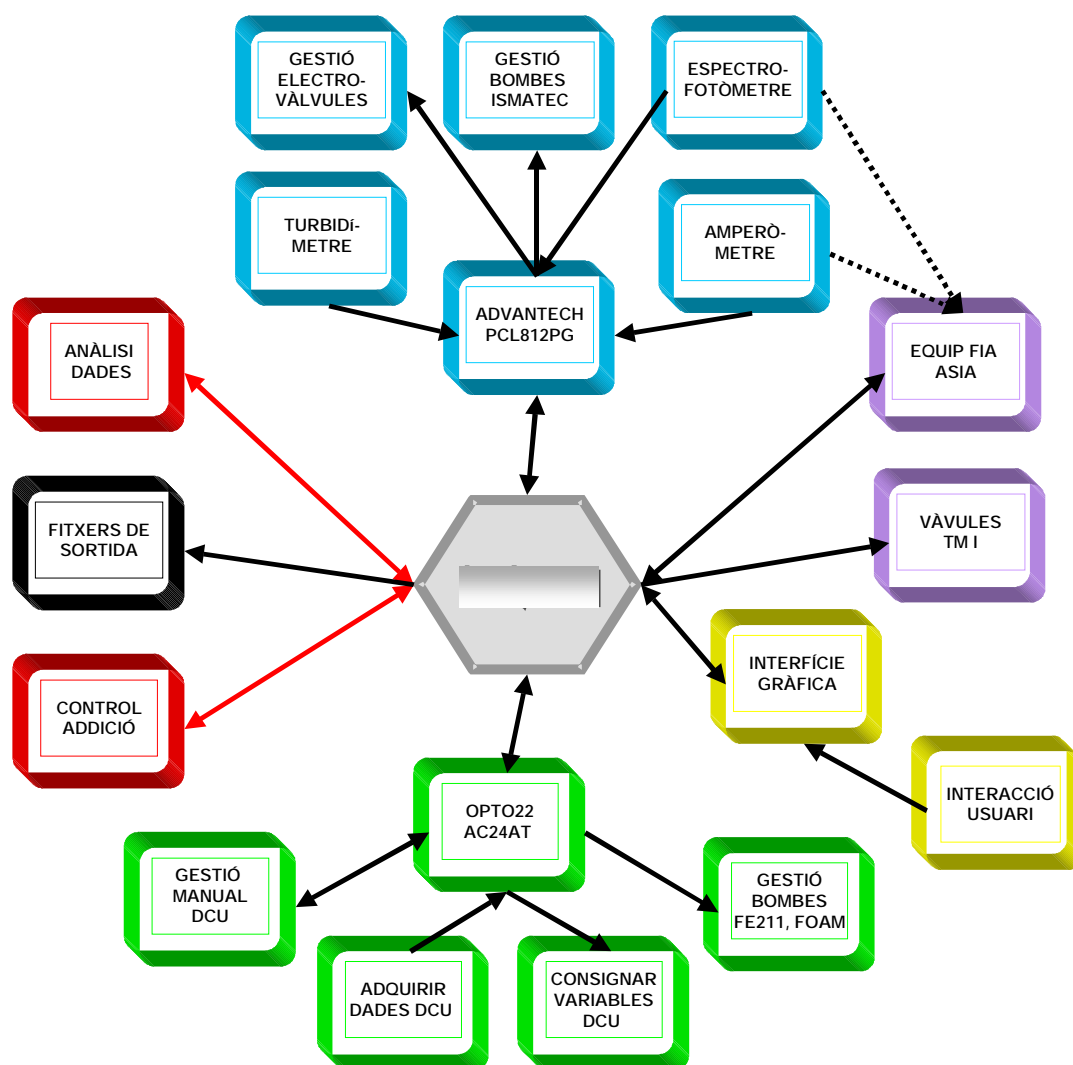


Figura 5.2. Diagrama de context de l'aplicació Sequen desenvolupada en aquest treball. En vermell es destaquen les parts que s'han dissenyat i adaptat específicament en funció de les característiques del sistema experimental que s'ha anat desenvolupant i modificant al llarg del treball. La resta dels mòduls no tenen una configuració variable en funció de les estratègies experimentals concretes. Les línies discontinúes són alternatives possibles d'adquirir els senyals d'aquests instruments.

Les diferents funcions que realitza *Sequen.prj* s'amplien al diagrama de context presentat a la Figura 5.2. Els mòduls de comunicació desenvolupats per aquest treball, com el *PCL\_lib*, *ASIA\_lib*, *TMI\_lib* i *DCU\_lib*, permeten la transferència de dades i/o ordres entre els diferents elements, les targetes de comunicació/ports sèrie i l'ordinador. La targeta d'adquisició de dades Advantech PCL-812-PG permet adquirir els senyals analògics de l'espectrofotòmetre, l'amperòmetre i el turbidímetre, així com comandar remotament les bombes Ismatec i les electrovàlvules Sirai des de l'ordinador, mitjançant el mòdul *PCL\_lib* i els respectius mòduls que gestionen conjuntament els diferents senyals. El mòdul *DCU\_lib* realitza funcions anàlogues sobre la placa Opto22 AC24AT (port sèrie RS-422-A), permetent així actuar sobre la DCU del Biostat MCD; adquireix els valors de les variables designades, consigna els valors dels diferents controladors, podent canviar els seus paràmetres, comanda les diferents bombes de l'equip o permet fer accions puntuals de forma manual (a l'apartat 8.3 es descriuen amb més detall les targetes utilitzades). Tant l'equip d'anàlisi FIA comercial (ASIA), com l'equip de vàlvules per mòduls (TMI) es comanden remotament amb els mòduls *ASIA\_lib* (port sèrie RS-232-C) i *TMI\_lib* (port sèrie RS-485) respectivament. En la confecció d'aquest treball no s'ha utilitzat l'equipament FIA, necessari per realitzar l'anàlisi de substrats com la glucosa i el lactat, ja que tant el seu desenvolupament com la seva aplicació s'han dut a terme en un altre treball experimental del mateix grup de recerca. Per tant, en l'apartat de la descripció detallada dels mòduls de comunicacions no es fa cap ressenya als mòduls relacionats amb l'equipament d'anàlisi de substrats per tècnica FIA (*ASIA\_lib*, *TMI\_lib*, *Amp\_lib* i *Fot\_lib*), que es recullen a la corresponent memòria del treball esmentat (Casablanca, 2000).

El mòdul *Proces* conté les accions de control d'addició, i l'anàlisi i el processament de dades. Els mòduls *Protocol* i *Sequen* juntament amb la subrutina d'interfície gràfica, *Sequen.uir*, permeten la presentació dels diferents senyals així com el comandament remot per a diferents elements mitjançant els seus panells. L'usuari interacciona amb aquesta interfície a través de menús, panells, caixes diàleg, caixes llistat, gràfics científics i sortides per impressora. La forma d'utilitzar l'aplicació *Sequen.prj* es defineix més endavant. Els mòduls *Proces* i *Protocol* també realitzen les sortides per fitxer de les dades processades i de les dades adquirides de la diferent instrumentació.

### **5.5.1. Mòduls de comunicacions.**

#### **5.5.1.1. Mòdul *DCU\_lib*: controlador de la DCU.**

El mòdul *DCU\_lib* fa d'interpret en la comunicació que s'estableix via port sèrie RS-422-A (Opto22 AC24AT) entre l'ordinador de gestió i la unitat de control i mesura digital (DCU) del Biostat MCD. Les dades entre els dos equips s'envien mitjançant un protocol

Mestre/Esclau (*Master/Slave*), és a dir, l'ordinador fa una petició de dades a la DCU i només llavors la DCU envia les dades demanades a l'ordinador de gestió. Les dades s'envien com paquets de 10 bits (1 d'inici, 7 de dades, 1 de paritat i 1 de parada) a una velocitat de 9600 bauds. Els 7 bits de dades es llegeixen com un caràcter comprès entre els 128 primers valors de la taula ASCII. La fiabilitat de la recepció de dades s'assegura amb la presència del bit paritat (paritat senar) per a cada paquet de dades i una comprovació (*checksum*) de tot el conjunt de dades enviades. Cada conjunt de dades acaba amb el valor de comprovació calculat a partir de la suma dels valors ASCII de cada caràcter enviat. Alguns aspectes crítics en la comunicació són l'interval mínim entre dos demandes consecutives de dades (0.5 segons) i la necessitat d'enviar un senyal d'interrupció (*break*) a l'inici de les comunicacions per netejar la línia de possibles restes d'una comunicació establerta anteriorment.

El mòdul DCU\_lib proporciona a l'aplicació *Sequen.prj* els valors de les variables del procés com el pH, la temperatura, la velocitat d'agitació, l'oxigen dissolt, els consums de CO<sub>2</sub>, àlcali i antiescumejant, etc., així com les consignes i l'estat dels controladors (p. ex. TEMP, STIRR, PH i PO2) de les variables més importants del procés. Aquest mòdul permet enviar nous valors de consigna als controladors, com també l'activació o desactivació remota d'aquests. També pot enviar ordres de nous cabals d'addició de substrat a la bomba FE-211 i/o comandar la bomba FOAM On/Off des de l'ordinador. Encara que la DCU gestiona els mòduls de control del Biostat MCD, en cas que fos necessari es podria fer el control d'alguna de les variables del procés amb un anell de control extern gestionat des de l'ordinador. També es poden canviar els paràmetres dels mòduls de control de la DCU des del programari de control del procés. Algunes de les funcions interessants del mòdul DCU\_lib són:

- **DCU Inicialitza**: S'inicialitza la comunicació amb la DCU. Obre el port de comunicació en COM2 (2F8), el configura amb els paràmetres descrits anteriorment i fixa la DCU per operar en mode remot (*Remote*). Al laboratori es disposa de dos equips Biostat MCD cadascun d'ells amb una DCU i estació de mescla gasos diferents. Tot i que funcionen de forma molt similar contenen, però, un nombre diferent de variables i d'anells de control del procés. Per tant, prèviament a qualsevol acció, s'ha de definir el tipus de DCU amb què es treballa (les assignacions de cadascuna de les variables de cada Biostat MCD i el tipus de DCU es mostren a l'*Apèndix A*). Durant la inicialització també se sol·liciten a la DCU els valors actuals per a tots els punts de consigna - *set-points* (CS), els modes del controlador (CM), les entrades digitals (DI) i les variables del procés (PV).
- **DCU EnviaRebVerifica**: Amb aquesta funció es pot enviar una ordre directament, és a dir, es pot enviar una cadena de caràcters (*string*) que correspon al format que entén la DCU, per exemple l'ordre DS:0:0:BC:1:1 és la cadena que s'utilitza per posar la DCU en operació remota i així poder actuar sobre la DCU a distància. La funció



\*DCU\_ArreglaOrdre s'encarrega de donar-li el format que ha de tenir incloent el caràcter d'inici de telegrama (\*), l'identificador de la DCU (1 o 2), el valor de comprovació final i el caràcter de final de telegrama (@), de forma que el telegrama que s'obtidria quedaria finalment com \*1:DS:0:0:BC:1:1:975@. La resposta de la DCU a aquest telegrama ha de ser #1:DS:0:17:619@. De fet qui realment envia el telegrama al port sèrie és la funció DCU\_EnviaReb, que també recull la resposta. Després de rebre la resposta, la funció DCU\_EnviaReb torna la resposta a DCU\_EnviaRebVerifica. Aquesta verifica si el valor de comprovació del telegrama resposta coincideix amb el valor calculat i enviat per la DCU. La funció efectua en total dos intents per obtenir una resposta correcta, i en cas contrari surt d'aquesta i proporciona un error pel no establiment de la comunicació.

- DCU\_LlegeixPVs: Amb aquesta funció, s'obté un punter al vector que conté els valors de totes les variables del procés sol·licitades en una petició de dades. Primer s'ha d'enviar un telegrama que, arreglat, té la forma \*1:SU:0:0:PV:5:2:1030@. D'aquesta forma se li diu a la DCU quins tipus de variables es volen adquirir, en aquest cas serien les variables de procés (PV) i la DCU ha de respondre si la comunicació s'ha efectuat correctament enviant #1:SU:0:17:636@. Una vegada s'ha configurat correctament els tipus de variables a la DCU, tot seguit s'envia l'ordre per adquirir les dades, amb el telegrama \*1:DR:0:0:569@. Si la DCU rep correctament l'ordre envia una resposta amb els valors de les variables sol·licitades. Un exemple de resposta podria ser la seqüència de caràcters: #1:DR:0:17:PV:1:157.0:2:163.0:3:0.0:4:0.0:5:1368.0:6:1097.6:7:64.8:8:0.0:9:0.0:10:0.2:11:1.2:12:998.8:13:0.2:14:0.2:15:1389.9:16:0.0:17:0.0:7432@. Després de rebre el telegrama resposta, la funció DCU\_EnviaRebVerifica comprova que el valor de comprovació de la resposta coincideix amb el valor calculat i enviat per la DCU (per l'exemple hauria de ser 7432). Seguidament s'analitza la resposta, convertint la cadena dels caràcters rebuts en valors numèrics intel·ligibles per a l'usuari. Se segmenta la resposta (el caràcter “:” indica la separació de camps i cada variable té un número assignat), obtenint-se valors com 157.0 per a la variable temperatura i 1368 per a la de l'oxigen dissolt. Més exemples de telegrams, així com un llistat dels valors dels guanys i de les compensacions de cada variable en cada tipus de DCU es mostren a l'Apèndix A.
- DCU\_LlegeixPV: Llegeix el valor d'una única variable del procés si es demana en l'aplicació. Se li pot indicar si es vol el valor de la última lectura, o bé si es desitja forçar a la DCU a què torni a prendre una mesura de la variable.

De forma bastant similar s'opera amb la resta de funcions anàlogues a DCU\_LlegeixPVs i DCU\_LlegeixPV canviant en cada cas l'ordre enviada en funció del tipus de variables requerida. Aquestes poden ser els valors actuals per a tots els punts de consigna (CS), els valors de l'estat del controlador (CM), o els valors de les entrades digitals (DI). Si es

volen gestionar remotament els controladors de la DCU des de l'ordinador, existeixen dins del mateix mòdul funcions que ho permeten fer com DCU\_EscriuCM i DCU\_EscriuCS.

#### 5.5.1.2. Mòdul PCL\_lib: controlador de la targeta d'adquisició de dades.

El mòdul PCL\_lib gestiona els recursos analògics (A) i digitals (D) de la targeta d'adquisició de dades PCL812PG d'Advantech, és a dir, compta, ocupa i/o allibera recursos, inicialitza, fixa i torna els guanyos binaris i decimals dels diferents canals A/D, així com també realitza les conversions d'entrades A/D i de sortides D/A. Algunes de les funcions més rellevants d'aquest mòdul són:

- PCL\_Inicialitza: Inicialitza la targeta d'adquisició de dades. En cas de fer-se una reinicialització completa, també s'inicialitzen totes les estructures globals que contenen la informació dels usuaris de cadascun dels recursos de la targeta (entrades i sortides analògiques i digitals).
- PCL\_LlegeixUnRecursAnalogic: Aquesta funció s'utilitza per llegir una de les entrades analògiques de les que disposa la targeta d'adquisició, o bé per tornar el valor memoritzat d'una de les sortides analògiques, deixant en un punter a variable el valor recollit (entre 0.0 i 10.0 V).
- PCL\_LlegeixUnRecursDigital: S'utilitza aquesta funció per llegir una de les entrades digitals de les que disposa la targeta d'adquisició, o bé per tornar el valor memoritzat d'una de les sortides digitals, deixant en un punter a variable el valor recollit (entre 0 i 5.0 V).
- PCL\_EscriuUnaSortidaAnalogicaEnBinari: Amb aquesta funció es fixa el valor d'una sortida analògica, indicant el nivell de sortida mitjançant un valor sencer entre 0 i el valor màxim que depèn de la resolució (en aquest cas els 12 bits ens permeten un valor entre 0 i 4096). La tensió obtinguda a la sortida està entre 0.0 i 5.0 V.
- PCL\_EscriuUnaSortidaDigital: Aquesta funció escriu una sortida digital de la targeta d'adquisició de dades. La tensió obtinguda a la sortida està entre 0.0 i 5.0 V.

#### 5.5.1.3. Mòdul Bom\_lib: controlador de les bombes.

L'aplicació ha de gestionar diverses bombes analògiques. Aquestes bombes són de velocitat variable, podent-se canviar la velocitat, el mode de control i el sentit. Això implica l'enviament de senyals diverses. Per tant, resulta més eficaç que tot el tractament de les

bombes es faci en un mòdul separat, en lloc de treballar directament sobre la targeta d'adquisició de dades, i que siguin les funcions específiques de les bombes les que utilitzin les funcions de la targeta d'adquisició. D'aquesta forma sorgeix la necessitat de crear un mòdul propi per a les bombes peristàltiques Ismatec: Bom\_lib. Algunes de les funcions més rellevants d'aquest mòdul són:

- **BOM\_Inicialitza**: Amb aquesta funció s'inicialitza el controlador de les bombes, tot indicant (si es realitza una reinicialització) que són controlades remotament i fixant el sentit i la velocitat de treball per defecte.
- **BOM\_SetBomba**: Aquesta funció s'utilitza per consignar un dels paràmetres de la bomba, tot indicant sobre quin número de bomba i quin paràmetre d'aquesta es vol actuar:
  - ✓ Sentit (Sortida Digital a 0 = Horari o 1 = Antihorari).
  - ✓ Stop (1 = Parar la bomba i 0 = Posar en marxa).
  - ✓ Local (0 = Control remot i 1 = Control en bomba).
  - ✓ Velocitat Externa/Interna.
  - ✓ Velocitat de la bomba (Sortida analògica).
  - ✓ Velocitat real de la bomba (Entrada analògica).
- **BOM\_SetBombaString**: Amb aquesta funció, s'envia una ordre que fixa un dels paràmetres a un valor determinat, en aquest cas un punter a cadena de caràcters amb el següent format: "NúmeroBomba:Paràmetre:Valor", on paràmetre pot ser: SENTIT, STOP, LOCAL, V\_EXTERN o SPEED (el significat d'aquests paràmetres es descriuen amb més detall a l'apartat 5.5.7.1).

#### 5.5.1.4. Mòdul ELV\_lib: controlador de les electrovàlvules.

L'aplicació ha de gestionar varies electrovàlvules digitals. Aquestes necessiten 12 V de tensió per commutar, en canvi la sortida digital de la targeta d'adquisició de dades només ho fa en un valor màxim de tensió de 5.0 V. Això implica que en una connexió directa quan s'envii un senyal des de l'ordinador per obrir l'obturador d'una de les electrovàlvules no ho podrà fer. Per tant, per solucionar aquesta problemàtica s'utilitza una targeta de control de relés amb connexions TTL (*Transistor-Transistor-Logic*), amb una font d'alimentació de 13.8 V (a l'apartat 8.3.3 es proporciona més informació sobre aquesta targeta i la caixa de comunicacions dissenyada).

De la mateixa forma que en el mòdul BOM\_lib, al mòdul ELV\_lib el funcionament de les electrovàlvules s'ha encapsulat, essent les seves funcions les mateixes que utilitzen les

funcions del mòdul de la targeta d'adquisició: PCL\_lib. Algunes de les funcions més rellevants d'aquest mòdul són:

- **ELV\_Inicialitza**: Amb aquesta funció s'inicialitza el mòdul de les electrovàlvules. Si fa una reinicialització, reinicialitza totes les electrovàlvules i allibera les línies ocupades.
- **ELV\_SetElectrovalvula**: Aquesta funció s'utilitza per consignar un dels paràmetres de l'electrovàlvula, tot indicant sobre quin número d'electrovàlvula es vol actuar i el paràmetre de ON (1 = Oberta i 0 = Tancada).
- **ELV\_SetElectrovalvulaString**: Amb aquesta funció, s'envia una ordre que canvia l'estat d'una electrovàlvula donada a un nou valor, en aquest cas un punter a cadena de caràcters amb el següent format: "NúmeroElectrovalvula:Paràmetre:Valor", on paràmetre pot ser només ON.
- **ELV\_GetElectrovalvula**: Amb aquesta funció, es crida a la funció de la targeta d'adquisició de dades PCL\_LlegeixUnRecursDigital per conèixer en quin estat es troba l'electrovàlvula consultada.

#### 5.5.1.5. Mòdul BIO\_lib: controlador del turbidímetre.

Com ja s'ha comentat, la targeta és de 12 bits el que vol dir que cadascun dels 4096 llocs en què se secciona el rang de 0 a 10 V del turbidímetre correspon a un valor de resolució de 0.025%. De la mateixa forma que en el mòdul BOM\_lib, al mòdul BIO\_lib s'ha encapsulat el funcionament del turbidímetre, essent les seves funcions les que utilitzen les funcions del mòdul de la targeta d'adquisició: PCL\_lib. Algunes de les funcions més rellevants d'aquest mòdul són:

- **BIO\_Inicialitza**: Aquesta funció inicialitza el controlador del turbidímetre, indicant al controlador de la targeta quins són els recursos que utilitza el sensor de biomassa (de la mateixa forma que ho fan cadascun dels controladors basats en la targeta d'adquisició de dades), i fixant quin és el guany i el desplaçament corresponent al sensor, per ser utilitzats en el moment de la lectura.
- **BIO\_LlegeixBiomasa**: Amb aquesta funció s'obté el nivell de terbolesa en tant per cent (%), deixant el seu valor en un punter a variable.
- **BIO\_Shutdown**: Allibera el recurs de la targeta d'adquisició que estava ocupant el turbidímetre.

## 5.5.2. Mòduls de control lògic de l'aplicació.

### 5.5.2.1. Mòdul Sequen.

El mòdul Sequen desenvolupa dues funcions fonamentals en l'execució del control lògic de l'aplicació. La primera és fer de pont d'enllaç entre els mòduls de la interfície gràfica (funcions pròpies de la biblioteca *Interface Library* de LabWindows/CVI) i el codi dels diferents mòduls. La segona és portar a terme la inicialització pròpiament dita de l'aplicació. La seqüència d'instruccions més importants que es descriuen en la funció principal són:

- S'instal·la la crida principal del programa, la funció PTC\_ProcessLoop. Cada cop que hi ha un èxit (*event*) aquesta funció es posa en marxa. La funció SetIdleEventRate( ) fixa el ritme en què té lloc el succés desocupat (*Idle*) essent de 100 ms.
- Es carreguen els diferents panells, es consignen controls i s'actualitzen les dades en les caselles i botons gràfics pels diferents equipaments que envien senyals (p. ex. la DCU i el turbidímetre d'Aquasant).
- Es carrega automàticament l'arxiu de protocol INICI.TXT i s'executen les ordres que tingui definides en la caixa llistat d'ordres (normalment són instruccions per inicialitzar les comunicacions dels diferents equips, així com el directori per defecte on es guarden o busquen els fitxers que produeixi o necessiti l'aplicació, veure la Figura 5.3).



Figura 5.3. Panell Protocol de l'aplicació Sequen (caixa llistat d'ordres) amb l'arxiu INICI.TXT carregat.

- Es queda l'aplicació en marxa a l'espera que es carregui en la caixa llistat d'ordres (panell Protocol) de l'aplicació Sequen un arxiu de protocol per executar.

#### 5.5.2.2. Mòdul Protocol.

El mòdul Protocol exerceix la funció més important de l'aplicació **Sequen**: la d'interpretar i executar les ordres definides en l'arxiu que s'ha carregat en la caixa llistat d'ordres. Aquesta tasca l'acompleix sobretot gràcies a tres funcions que es troben definides en ella, les quals es descriuen breument a continuació:

- **PTC ProceLoop**: Aquesta funció primer actualitza el temps i pregunta si cal processar alguna ordre de l'arxiu de protocol. Si el protocol està iniciat seguidament entra a executar les ordres, preguntant si ja ha executat l'ordre sobre la que es troba la marca (*checklist*). Si no ho ha fet, crida la funció **PTC\_ExecutaOrdre** i si ja l'ha llegida, crida la funció **PTC\_LlegeixSeguentLinia**. Seguidament, s'actualitzen les mesures pels diferents equipaments que envien senyals (p. ex. la DCU amb la funció **PTC\_ProcessaMesuresDCU** i el turbidímetre amb la funció **PTC\_ProcessaMesuresBIO**).
- **PTC ExecutaOrdre**: Aquesta funció és el veritable executor i intèrpret d'ordres dels arxius de protocol (\*.txt) escrits amb el llenguatge propi de l'aplicació: la sintaxi d'ordres **Sequen** (especificacions de les diferents paraules clau de la sintaxi es descriuen a l'apartat 5.6). Els passos bàsics que segueix la funció són: primer mira el temps i si ha arribat el moment d'interpretar l'ordre ho fa i després l'executa. Quan ja ha interpretat l'ordre de la caixa llistat d'ordres llegeix la següent línia. Això ho fa fins que passa un dels següents esdeveniments: 1) s'arriba al final de l'arxiu de protocol (EOF); 2) es llegeix una ordre que encara no s'ha d'executar, és a dir, el programa s'ha d'esperar un temps. En aquesta funció estan definides totes les expressions pròpies de la sintaxi d'ordres **Sequen**.
- **PTC LlegeixSeguentLinia**: Aquesta funció efectua els salts de línia de les ordres contingudes en la caixa llistat d'ordres. Primer de tot mira l'ordre i intenta saltar fins la línia següent de l'arxiu del protocol. Si ha pogut arribar llegeix l'ordre següent.

Altres funcions d'aquest mòdul a les quals cal fer esment són les del processament de les mesures provinents dels equips externs. Com exemple es mostra la de la DCU, però s'ha de tenir en compte que per altres mesures s'opera de forma similar.

- **PTC ProcessaMesuresDCU**: Aquesta funció opera de la següent forma. Primer, mira si ha de fer alguna mesura comparant el temps actual amb el que resta per fer la següent mesura. Seguidament actualitza el compte enrera de la mesura i mira si el port sèrie de comunicació amb la DCU està disponible sense interrogar directament al port. Adquireix les variables de la DCU (T, pH, pO<sub>2</sub>, V<sub>CO<sub>2</sub></sub> i V<sub>NaOH</sub>) i el temps, tot col·locant els nous

valors en el panell gràfic de la DCU. Registra aquests valors en un fitxer de sortida per a dades DCU. Per últim, actualitza els gràfics de pH i pO<sub>2</sub> de la finestra de la DCU.

Altres funcions que cal presentar són les funcions que permeten inserir ordres en l'arxiu de protocol ja carregat en la caixa llistat d'ordres i que quan es criden són introduïdes just després de la última ordre executada en l'arxiu de protocol. Aquestes funcions són imprescindibles quan s'hagin de fer accions de control i gestió sobre l'equipament del procés, ja que permeten inserir noves instruccions quan són requerides en el mòdul Proces. Es disposa de dues funcions:

- PTC\_AfegeixComentari: Funció per afegir comentaris a la caixa llistat d'ordres estant definida a Protocol.h. Té els següents paràmetres:
  - ✓ char \*text és una cadena (*string*) acabada en '\0' amb el text que s'ha de visualitzar.
  - ✓ int color és un dels següents valors definits a Protocol.h: PTC\_COLOR\_ORDRE, PTC\_COLOR\_ERROR, PTC\_COLOR\_COMENTARI.
- PTC\_ProtocolInsereixOrdre: Funció per afegir la cadena ordre a la caixa llistat d'ordres del protocol. L'ordre s'executa quan s'acaba de processar l'ordre actual. La funció torna PTC\_NO\_ERROR si ha pogut inserir-la i PTC\_OVERFLOW\_FIFO\_ORDRES en cas contrari.

### 5.5.2.3. Mòdul Proces.

Aquest mòdul conté la funció Processa() que alhora conté totes les funcions que permeten fer l'anàlisi i el processament de les dades adquirides pel programa durant el cultiu, així com la o les accions de control desenvolupades en el decurs d'aquest treball, i les sortides per fitxer de totes les mesures processades i estimades. La crida a aquest mòdul sempre es fa des de l'arxiu de protocol carregat en la caixa llistat d'ordres, executant-se l'ordre amb una sintaxi definida com: **PROC**, seguida de l'argument de la funció que es crida de la funció Processa()(p. ex. **PROC CALCULA OUR**).

Al principi d'aquest treball, totes les eines s'estaven desenvolupant (p. ex. les mesures en línia) i, per tant, aquest mòdul no contenia dins seu cap funció que pogués actuar amb una acció de control sobre el procés de cultiu i la seva instrumentació. A mesura que es van anar validant les diferents mesures, així com les diferents estratègies de control, aquest mòdul es va anar completant amb les funcions necessàries per portar a terme els objectius de control. Per tant, per seguir en aquesta memòria una estructura lògica i ordenada, les diferents funcions implementades es mostraran juntament amb els processos per als que es van desenvolupar.

### 5.5.3. Altres mòduls.

El mòdul Vars conté, dins seu, gran quantitat de variables definides i constants que s'utilitzen tant en el mòdul Proces com en el mòdul Sequen. DCU\_ctrl, Bom\_ctrl i Elv\_ctrl són mòduls que permeten l'enllaç entre els panells gràfics Control manual de la DCU (Figura 5.4), Control de les bombes i Control de les electrovàlvules (Figura 5.5), i el codi generat per poder interaccionar des de finestres amb aquest elements.



Figura 5.4. Panell gràfic Control manual de la DCU que està inclòs dintre de la finestra Seqüenciació d'ordres.

Un cas a banda és el mòdul Express. Aquest mòdul sorgeix de la necessitat de poder realitzar en l'arxiu de protocol operacions lògiques i aritmètiques senzilles amb les variables i expressions que més tard apareixeran carregades com ordres en la caixa llistat d'ordres. A grans trets, aquest mòdul proporciona un intèrpret més avançat d'expressions, tractant-les a fi de que el mòdul Protocol reconegui la instrucció que s'ha definit en l'arxiu de protocol carregat.

### 5.5.4. Interfície gràfica d'usuari.

El tipus d'interfície amb la que es presenta l'aplicació és de finestra, ja que Sequen s'ha desenvolupat en l'entorn LabWindows/CVI per a Microsoft Windows, utilitzant-se els controls proporcionats per LabWindows/CVI per al desenvolupament de panells frontals. Del gran nombre de controls disponibles en LabWindows/CVI, se'n utilitza un nombre reduït, amb l'objectiu de donar a l'aplicació un aspecte el més homogeni possible. Algunes de les



eines que s'han utilitzat en el disseny dels diferents panells són: barres de menú per escollir entre diverses alternatives (*Menu bars*), panells per a cada equip o instrument (*Panel*), icones per obrir i tancar panells (*Picture*), caixes llistat per poder seqüenciar les ordres (*List box*), botons per poder realitzar diferents accions, com inicialitzar un instrument o tancar un panell (*Command botons*), caselles numèriques o de text per mostrar un valor numèric, o bé un text (*Numeric i String*), quadres i missatges que mostren texts extensos (*Text box i Text message*), elements decoradors de les diferents eines (*Decoration*), gràfics per poder visualitzar les dades en temps real (*Strip chart*), dispositius lluminosos per saber l'estat del controlador (*LED*) o per saber el valor d'una mesura amb format tipus voltímetre (*Ring meter*), interruptors binaris per commutar una electrovàlvula o una bomba (*Binary switch*) o barres de selecció, amb diverses possibilitats (*Ring slide*). A les Figures 5.5 i 5.6 es mostren els resultats finals del disseny de l'aplicació amb alguns dels panells que s'han fet servir oberts.

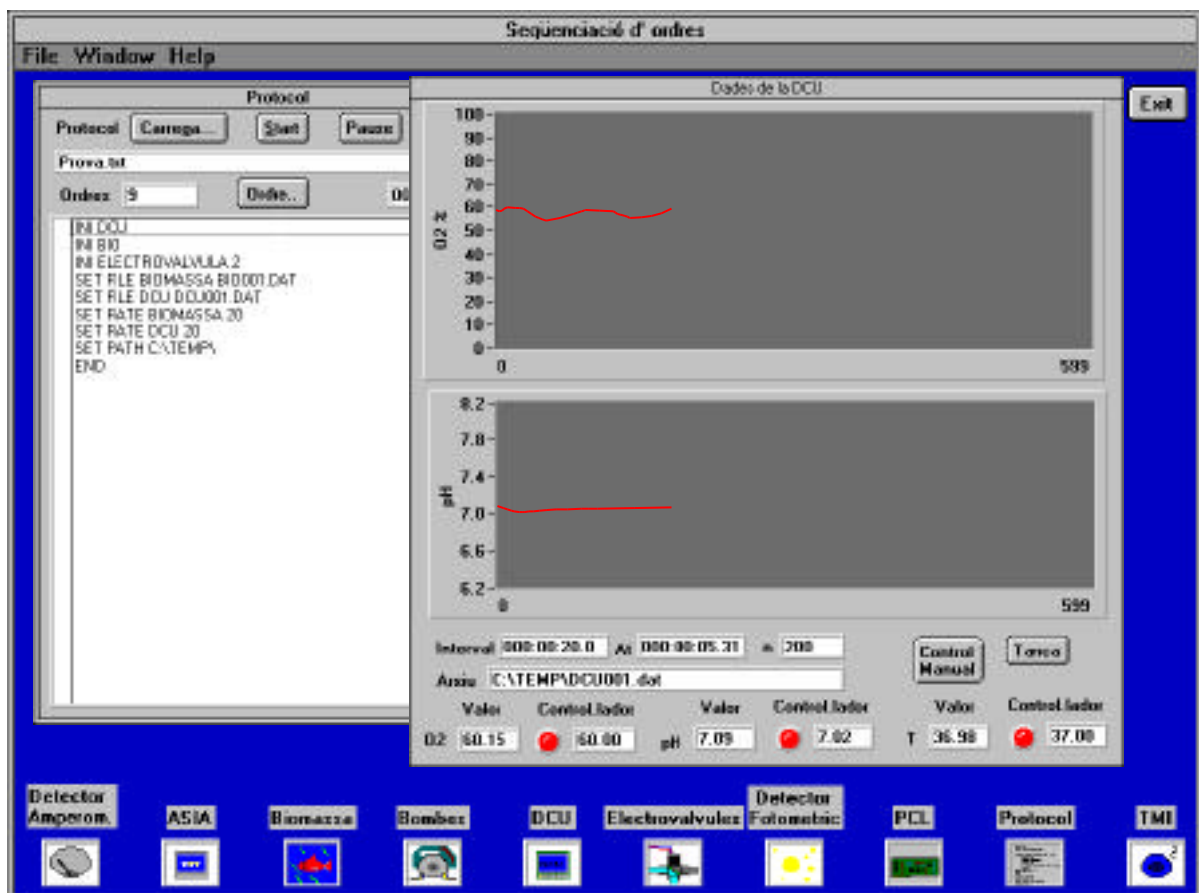


Figura 5.5. L'aplicació Sequen amb els panells de Protocol i Dades de la DCU oberts.

L'aplicació desenvolupada permet la interacció de l'usuari a partir d'una finestra única i diversos panells definits per a cada equip o instrument. La interfície gràfica d'usuari s'ha dissenyat per facilitar la lectura dels paràmetres del procés i la gestió remota eficient dels diferents elements. Així, es poden visualitzar i subministrar fàcilment paràmetres crítics del procés. La finestra Sequenciació d'ordres de Sequen.uir (\* .uir, extensió per a les interfícies

gràfiques), pot mostrar tots els panells alhora o un per un (Figura 5.5 i 5.6). Sota els gràfics per a cada variable del procés es mostra el temps que manca per a la següent mesura, així com el nombre de mesures realitzades fins aquell moment i la direcció de l'arxiu on estan registrades. La T, el pH, el pO<sub>2</sub> o la turbidimetria del cultiu es mostren de forma fàcil per a la seva lectura, tant en valors numèrics com en gràfics (es poden visualitzar fins a 600 dades de cop, quan exhaureix el límit escombra el gràfic de dades i incrementa el comptador). Els botons lluminosos (LEDs) indiquen clarament si està o no activat el controlador corresponent. El botó Inicialitza permet inicialitzar el port de comunicació al que està vinculat cada aparell. Fent un simple clic en el botó Tanca es tanca el panell i si es fa sobre una de les icones gràfiques aquest s'obre. També hi ha l'opció d'obrir i tancar panells amb la barra de menú Window. Les bombes i les electrovàlvules (Figura 5.6) poden ser gestionades remotament amb el canvi dels diferents paràmetres i la DCU també es pot comandar totalment enviant instruccions del tipus llenguatge DCU a través del panell Control manual de la DCU (Figura 5.4) que s'activa amb el botó Control Manual del panell Dades de la DCU.

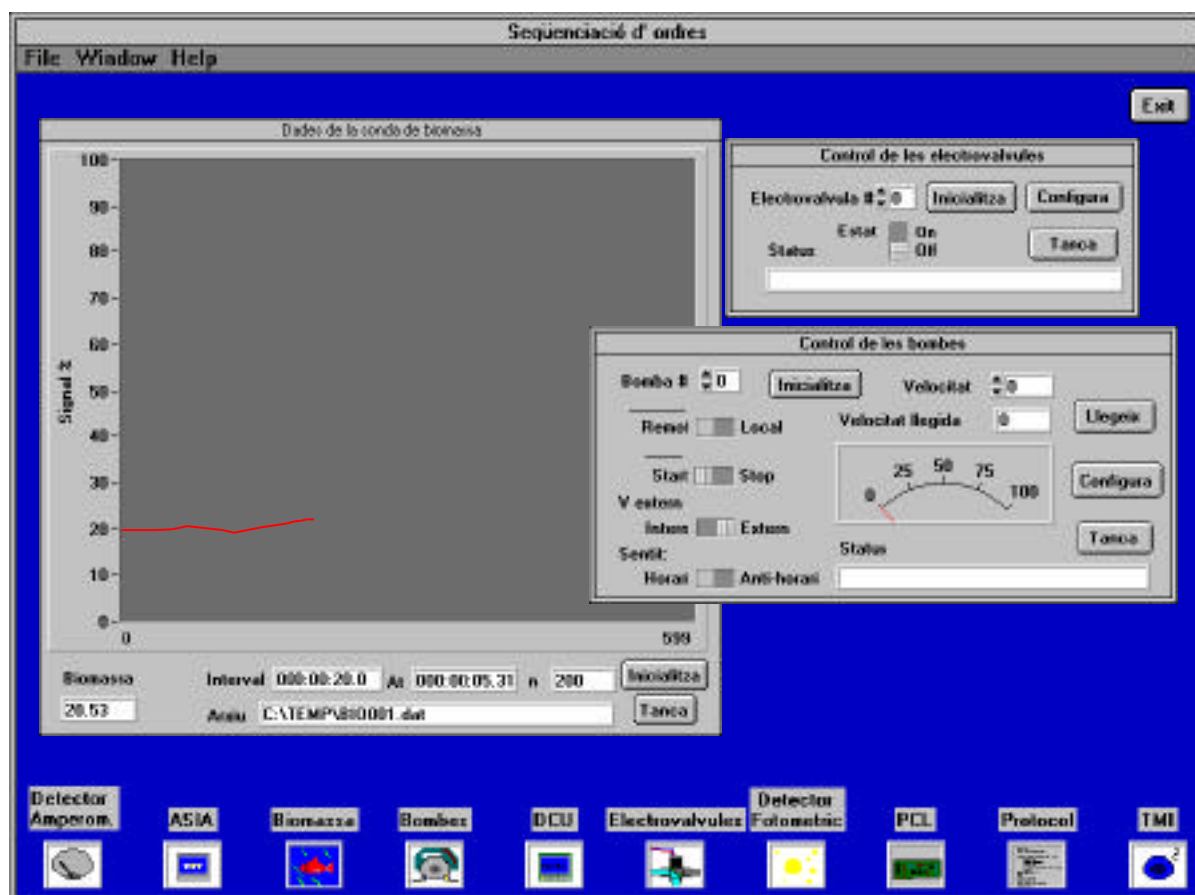


Figura 5.6. L'aplicació Sequen amb els panells oberts de Dades de la biomassa, Control de les bombes i Control de les electrovàlvules.

A la part superior dreta del panell Protocol (Figura 5.3) es mostra el temps global transcorregut pel procés. En aquest panell es poden carregar amb el botó Carrega els diferents

arxius de protocol que es vulguin executar, així com també es pot introduir, amb el botó Ordre, noves consignes de qualsevol variable predefinida (p. ex. \$DCU\_CS\_TEMP) o de nova definició, o bé qualsevol ordre que es vulgui inserir en la seqüència (sempre escrita d'acord la sintaxi de Sequen). El botó Start/Stop permet, un cop carregat l'arxiu de protocol, iniciar o aturar la seqüència d'ordres descrites en el fitxer. El botó Pause/Resume fa una aturada momentània de l'execució de l'arxiu de protocol. Per poder tornar a l'operació normal, només cal fer un clic un altre cop al botó.

Name	C	S	I	Date
amp_lib.c	☐	C		14/05/99, 22:10 a.m.
amp_lib.h	☐			14/05/99, 19:10 a.m.
asia_ctl.c	☐	C		01/04/99, 20:06 a.m.
asia_ctl.h	☐			28/03/99, 19:21 a.m.
asia_ctl.uir	☐			28/03/99, 22:36 a.m.
asia_lib.c	☐	C		24/04/99, 13:19 a.m.
asia_lib.h	☐			24/04/99, 10:01 a.m.
bio_lib.c	☐	C		01/04/99, 11:08 a.m.
bio_lib.h	☐			14/02/99, 21:08 a.m.
bom_ctrl.c	☐	C		01/04/99, 12:34 a.m.
bom_ctrl.h	☐			31/01/99, 19:31 a.m.
bom_ctrl.uir	☐			11/03/99, 18:50 a.m.
bom_lib.c	☐	C		01/04/99, 19:13 a.m.
bom_lib.h	☐			11/03/99, 18:48 a.m.
dcu_ctrl.c	☐	C		27/04/99, 0:00 a.m.
dcu_ctrl.h	☐			24/04/99, 17:00 a.m.
dcu_ctrl.uir	☐			24/04/99, 17:18 a.m.
dcu_lib.c	☐	C		14/06/99, 20:01 a.m.
dcu_lib.h	☐			01/02/0, 19:56 a.m.
elv_ctrl.c	☐	C		29/03/99, 21:49 a.m.
elv_ctrl.h	☐			02/02/99, 22:17 a.m.
elv_ctrl.uir	☐			10/02/99, 18:39 a.m.
elv_lib.c	☐	C		01/04/99, 11:08 a.m.
elv_lib.h	☐			14/02/99, 21:03 a.m.
expres.c	☐	C		27/05/99, 9:04 a.m.
expres.h	☐			03/02/0, 16:32 a.m.
fot_lib.c	☐	C		14/05/99, 22:20 a.m.
fot_lib.h	☐			14/05/99, 19:15 a.m.
pcl_lib.c	☐	C		10/04/99, 15:50 a.m.
pcl_lib.h	☐			10/04/99, 14:18 a.m.
proces.c	☐	C		10/02/0, 16:09 a.m.
proces.h	☐			10/02/0, 16:01 a.m.
protocol.c	☐	C		26/12/99, 23:50 a.m.
protocol.h	☐			09/02/0, 17:50 a.m.
sequen.c	☐	C		02/12/99, 23:19 a.m.
sequen.h	☐			26/12/99, 23:37 a.m.
sequen.uir	☐			21/09/99, 16:22 a.m.
tmi_ctrl.c	☐	C		01/04/99, 20:01 a.m.
tmi_ctrl.h	☐			31/01/99, 11:55 a.m.
tmi_ctrl.uir	☐			31/01/99, 12:29 a.m.
tmi_lib.c	☐	C		24/04/99, 13:18 a.m.
tmi_lib.h	☐			01/04/99, 20:02 a.m.
vars.c	☐	C		10/02/0, 14:14 a.m.
vars.h	☐			10/02/0, 14:14 a.m.

Figura 5.7. L'aplicació Sequen.prj amb tots els fitxers \*.c, \*.h i \*.uir.

L'aspecte normal de l'aplicació *Sequen.prj* abans de construir l'executable es presenta a la Figura 5.7, on es poden observar el total de fitxers \*.c, \*.h i \*.uir que s'han emprat en la realització d'aquesta aplicació. Una vegada es vol construir l'executable hi ha dues opcions: una opció és fer córrer l'aplicació *Sequen.prj* en el mateix entorn LabWindows/CVI amb el que l'entorn de programació no s'allibera fins que s'acaba l'execució del procés. L'altra opció consisteix en crear un executable de l'aplicació, *Sequen.exe* (per aquesta aplicació no arriben a ocupar més de 300 kB de memòria), de tal manera que es pot fer anar alhora tant el procés com l'entorn de programació. En resum, aquesta tasca ha requerit la programació de 38 arxius de codi en llenguatge C i LabWindows/CVI (arxius \*.c + \*.h) amb més de 15,000 línies que ocupen 767 kB de memòria.

### 5.5.5. Registre de dades.

Les sortides a fitxers s'han confeccionat pensant en possibles anàlisis i processaments posteriors de les dades adquirides en programes gràfics, fulls de càlcul, i processadors matemàtics i de textos. En la capçalera del fitxer sempre es defineixen la data (hora/dia/any) en què es va iniciar el recompte i el temps de procés transcorregut (hhh:mm:ss) des d'aquell moment. També hi han els noms de les variables que es registren. L'extensió del fitxer la defineix l'usuari (normalment \*.dat). Les dades de diferents columnes se separen per tabuladors, reservant la primera i segona columna respectivament al comptador de la variable (mostra) i al temps (hhh:mm:ss) en què ha estat adquirida. La informació de les diferents mesures es registra a la velocitat que defineix l'usuari en l'arxiu de protocol, essent la màxima velocitat a la que s'aconsella treballar de 2 registres/segon.

### 5.5.6. Errors durant l'execució.

Durant l'execució de l'aplicació es poden produir errors de diversos tipus (en general de comunicacions a l'inici de la seva configuració). En el cas de produir-se un error d'aquest o d'altre tipus, es mostra en el panell Protocol (caixa llistat d'ordres) una línia comentari de color vermell, tot just després d'executar-se l'ordre errònia, indicant a l'usuari el número de l'error, junt amb una descripció del mateix permetent així continuar en aquells casos en què l'error no sia important (p. ex. errors deguts a una adquisició dolenta d'un grup de dades de la DCU o variable no identificada, ja que no s'ha definit prèviament en l'arxiu de protocol). En el cas d'haver-hi errors en l'obertura/tancament d'algun arxiu, l'aplicació activa un quadre de diàleg (*Message popup*) per a què l'usuari es pugui assabentar de l'error i així corregir el problema detectat a posteriori. D'altra banda, quan l'aplicació *Sequen* està funcionant, es crea un fitxer, *Sequen.log*, que permet registrar tots els successos, ordres, accions i errors que s'han produït en l'aplicació durant l'execució de l'arxiu de protocol carregat per a la realització d'un determinat procés. De fet, aquest arxiu es comportaria com una veritable

“caixa negra” del procés, que es pot consultar a posteriori per esbrinar la causa dels errors no identificats i/o no detectats.

### 5.5.7. Sintaxi d'ordres Sequen.

Amb la sintaxi d'ordres Sequen es defineixen tota una sèrie d'instruccions que un cop interpretades pel mòdul Protocol, permeten interaccionar totalment amb els diferents mòduls de l'aplicació, i d'aquesta forma poder implementar i desenvolupar totes les tècniques i estratègies de control que aquest treball s'ha marcat com a objectius. La decisió de desenvolupar una sintaxi pròpia per a l'aplicació Sequen es basa en l'orientació donada al treball, en el sentit de posar a punt una eina que qualsevol usuari pugui fer servir d'una forma còmoda i senzilla, sense necessitat d'accedir contínuament al codi intern de l'aplicació. Les sentències i instruccions inicials de la sintaxi d'ordres Sequen es varen anar completant, a mida que va evolucionar el desenvolupament del treball, amb noves sentències i instruccions, quedant a la fi un llenguatge robust, fiable i flexible. La descripció de la sintaxi s'intenta fer agrupant totes aquelles ordres que tinguin característiques semblants o pròximes. La terminologia emprada per distingir ordres de la sintaxi i el caràcter espai (' '), es designen amb negreta i la lletra c respectivament (p. ex. **DCU<sub>c</sub>ORDRE**).

#### 5.5.7.1. Ordres per a la gestió d'equips.

- **BOMBA**: És una ordre per a les bombes, s'expressa de la següent forma **BOMBA<sub>c</sub>PARAMETRES<sub>c</sub>VALOR\_NOUS** i permet la gestió de totes les bombes peristàltiques Ismatec. Els *PARAMETRES* i *VALOR\_NOUS* es mostren a la Taula 5.2:

Valors de <i>PARAMETRES</i>	Rangs de <i>VALOR_NOUS</i>
<b>LOCAL</b>	0: remot 1: local
<b>SENTIT</b> (sentit de gir!)	0: horari 1: antihorari
<b>STOP</b>	0: start 1: stop
<b>V_EXTERN</b> (selector de velocitat!)	0: intern 1: extern
<b>SPEED</b>	número sencer entre 0 i 100, ambdós inclosos

Taula 5.2. Conjunt de paràmetres corresponents a les bombes amb els valors que poden adquirir.

- **DCU**: És una ordre per a la DCU de l'equip Biostat MCD, s'expressa amb la forma **DCU<sub>c</sub>ORDRE** i permet la gestió de tota la DCU. L'*ORDRE* que s'envia s'escriu segons la pròpia sintaxi de l'equip, però no cal ni l'índex d'enviament inicial (\*), ni l'acabament

final (@), ni tampoc el valor de comprovació (la forma d'enviar les ordres es tracta amb més detall a l'Apèndix A).

- **ELECTROVALVULA:** És una ordre per a les electrovàlvules, s'expressa amb la forma **ELECTROVALVULA***C***PARAMETRE***C***VALOR\_NOU** i permet la gestió de totes les electrovàlvules de 3-vies Sirai. El valors de *PARAMETRE* i *VALOR\_NOU* es mostren a la Taula 5.3:

Valors de <i>PARAMETRE</i>	Rangs de <i>VALOR_NOU</i>
<b>ON</b>	<b>0:</b> desactivat/tancat <b>1:</b> activat/obert

Taula 5.3. Paràmetre corresponent a les electrovàlvules amb els valors que pot adquirir.

#### 5.5.7.2. Ordres per inicialitzar dispositius (equips + detectors).

- **INI:** Aquesta ordre s'expressa amb la forma **INI***C***DISPOSITIUS***C***OPCIONES** i permet inicialitzar la gestió o adquisició via port sèrie i/o la targeta d'adquisició de dades de tots els equipaments i instruments dels quals es disposa. Els diferents *DISPOSITIUS* i *OPCIONES* es mostren a la Taula 5.4:

DISPOSITIUS	OPCIONES	VALOR
<b>BIOMASSA</b>	-	-
<b>BOMBA</b>	<i>VALOR</i> (número de bomba)	Del 0 al 3
<b>DCU</b>	-	-
<b>ELECTROVALVULA</b>	<i>VALOR</i> (número d'electrovàlvula)	Del 0 al 3

Taula 5.4. Conjunt de dispositius amb les opcions i els valors que poden adquirir.

#### 5.5.7.3. Ordres per a la gestió en l'adquisició de dades.

- **SET PATH:** Aquesta ordre s'expressa amb la forma **SET***C***PATH***C***PATH\_NOU** i permet fixar el nom del directori on es guarden els arxius de sortida i el Sequen.log, així com modificar la ruta d'accés (*path*) per defecte. *PATH\_NOU* és el nom complet del directori on es vol guardar la informació de l'arxiu de protocol.

- **SET FILE:** Aquesta ordre s'expressa amb la forma **SETcFILEcDISPOSITIUcARXIU** i permet l'adquisició de dades dels diferents equipaments i instruments de què es disposa, així com un canvi de l'arxiu de sortida. Els diferents *DISPOSITIUS* que es poden prendre mesures són **BIOMASSA** i **DCU** i *ARXIU* és el nom del fitxer on es vol guardar la informació, normalment acabat en *\*.DAT* o *\*.TXT*.
- **SET RATE:** Aquesta ordre s'expressa com **SETcRATEcDISPOSITIUScTEMPS** i permet fixar la velocitat en l'adquisició de dades dels diferents equipaments i instruments de què es disposa, així com modificar el ritme de l'adquisició de dades en l'arxiu de sortida. Els diferents *DISPOSITIUS* que es poden prendre mesures són **BIOMASSA** i **DCU** i *TEMPS* és la velocitat de registre al fitxer on es vol guardar la informació. Aquest valor *TEMPS* pot ser: un nombre real (s), dos nombres reals separats per ':' (m:s), o tres nombres reals separats per ':' (h:m:s). Si el temps és un nombre positiu, l'adquisició s'engega al ritme especificat, si el valor és negatiu, p. ex. -1, l'adquisició s'atura.
- **INC FILE:** Aquesta ordre s'expressa amb la forma **INCcFILEcDISPOSITIUS** i permet augmentar l'índex en l'arxiu de sortida fixat per l'ordre SET FILE. Els diferents *DISPOSITIUS* que es poden prendre mesures són **BIOMASSA**, **DCU**, **TOTS** (significa un increment de tots els arxius de sortida que estan definits en l'arxiu de protocol).
- **AUTOZERO:** Aquesta ordre s'expressa amb la forma **AUTOZEROcDISPOSITIUS** i permet fer el zero de la senyal de l'amperòmetre i l'espectrofotòmetre. Els diferents *DISPOSITIUS* sobre els que es pot fer aquesta acció són **AMP**, **FOT**.

#### 5.5.7.4. Etiquetes i sentències d'operació amb arxius de protocol i subrutines.

- **END:** És una etiqueta de final d'arxiu. **END** es col·loca per indicar final de l'arxiu de protocol. No cal etiqueta d'inici d'arxiu i no cal l'etiqueta de final en tots el arxiu.
- **LABEL-RETURN:** Són etiquetes de l'inici i el final de la subrutina respectivament. **LABELcETIQUETA** es col·loca sempre a l'inici i permet que es pugui identificar la subrutina que es crida des de l'arxiu de protocol. *ETIQUETA* és el nom que es dona a la subrutina. **RETURN** s'ha de col·locar sempre al final de totes les ordres de la subrutina per a què es retorni a l'arxiu d'on prové la crida.
- **GOSUB:** Aquesta ordre s'expressa amb la forma **GOSUBcETIQUETA** i permet fer un salt a una subrutina dins del mateix arxiu de protocol, però quan s'acaben d'executar les ordres

en aquest retorna a l'arxiu del que procedeix la crida. *ETIQUETA* és el nom que es dona a la subrutina a la que salta l'aplicació.

- **CALL:** Aquesta ordre s'expressa amb la forma **CALL***cETIQUETA* i permet fer un salt dins del mateix arxiu de protocol. *ETIQUETA* és el nom que es dona a la subrutina a la que salta l'aplicació.
- **GOSUB***cFILE:* Aquesta ordre s'expressa amb la forma **GOSUB***cFILEcPROTOCOL\_NOU* i permet fer un salt a un altre arxiu de protocol però quan s'acaben d'executar les ordres en aquest retorna a l'arxiu del que procedeix la crida. *PROTOCOL\_NOU* és el nom de l'arxiu de protocol complet (amb extensió) al que salta l'aplicació.
- **CALL FILE:** Aquesta ordre s'expressa amb la forma **CALL***cFILEcPROTOCOL\_NOU* i permet fer un salt a un altre arxiu de protocol. *PROTOCOL\_NOU* és el nom de l'arxiu de protocol complet (amb extensió) al que salta l'aplicació.

#### 5.5.7.5. Sentències de decisió lògiques.

- **REP( . . . )-END REP:** Aquesta estructura de decisió lògica (iteració numèrica d'ordres) s'expressa de la forma que es presenta tot seguit. *EXPRESSIO* pot ser qualsevol expressió numèrica o variable definida. S'executa la *ORDRE(S)* fins que el número d'iteracions assoleix el valor numèric de *EXPRESSIO*.

```

REPcEXPRESSIO
    ORDRE(S)
ENDcREP
    
```

- **IF( . . . )-ELSE-END IF:** Aquesta estructura de decisió lògica (salt condicional d'ordres) s'expressa de la forma que es presenta a continuació. *EXPRESSIO* pot ser qualsevol expressió que utilitzi operadors lògics. Si el valor de *EXPRESSIO* és cert, s'executa les *ORDRE(S)\_CERT* i si és fals, s'executa les *ORDRE(S)\_FALS*.

```

IFcEXPRESSIO
    ORDRE(S)_CERT
ELSE
    ORDRE(S)_FALS
ENDcIF
    
```

- **WHILE( . . . )-END WHILE:** Aquesta estructura de decisió lògica (iteració condicional d'ordres) s'expressa de la forma que es presenta tot seguit. *EXPRESSIO* pot ser qualsevol



expressió numèrica o lògica. S'executa la *ORDRE(S)* mentre que el valor *EXPRESSIO* és cert.

```
WHILECEXPRESSIO  
    ORDRE(S)  
ENDCWHILE
```

- **BREAK**: És una sentència per sortir d'una iteració, és a dir, **BREAK** fa abandonar la iteració (**REP / WHILE**) actual. Salta (i executa) el corresponent **END (REP / WHILE)**.

#### 5.5.7.6. Ordres de l'arxiu de registre (Sequen.log).

- **LOG OFF**: Aquesta ordre s'expressa amb la forma **LOG***C***OFF** i permet parar el registre d' ordres de l'arxiu Sequen.log.
- **LOG ON**: Aquesta ordre s'expressa amb la forma **LOG***C***ON** i permet tornar a engegar el registre d' ordres de l'arxiu Sequen.log.
- **LOG FLUSH**: Aquesta ordre s'expressa amb la forma **LOG***C***FLUSH** i permet buidar l'arxiu de registre d'ordres. L'estat no canvia, i per tant si estava enregistrant segueix fent-ho.

#### 5.5.7.7. Ordres amb la variable temps.

- **DELAY**: Aquesta ordre s'expressa amb la forma **DELAY***C**TEMPS* i permet el retard en la seqüència d'ordres de l'arxiu de protocol. Aquest valor *TEMPS* pot ser: un nombre real (s), dos nombres reals separats per ':' (m:s), o tres nombres reals separats per ':' (h:m:s).
- **SET TIME**: Aquesta ordre s'expressa amb la forma **SET***C***TIME***C**TEMPS* i permet fixar el valor de la variable en l'arxiu de protocol. Aquest valor *TEMPS* pot ser: un nombre real (s), dos nombres reals separats per ':' (m:s), o tres nombres reals separats per ':' (h:m:s).
- **INC TIME**: Aquesta ordre s'expressa amb la forma **INC***C***TIME***C**TEMPS* i permet l'increment de la variable temps fixada prèviament en l'arxiu de protocol. Aquest valor *TEMPS* pot ser: un nombre real (s), dos nombres reals separats per ':' (m:s), o tres nombres reals separats per ':' (h:m:s).
- **WAIT TIME**: Aquesta ordre s'expressa amb la forma **WAIT***C***TIME** i permet a l'arxiu de protocol s'esperar fins que el temps d'execució és major o igual que la variable temps.

#### 5.5.7.8. Crida a Processa().

- **PROC**: Aquesta ordre s'expressa amb la forma **PROC***C***ARGUMENT** i crida a la funció Processa() del mòdul Proces. Aquesta funció a la seva vegada conté definides les diferents funcions que s'han implementat per fer l'anàlisi i el processament de les dades adquirides durant el cultiu i el control del procés. El *ARGUMENT* és el nom que defineix a les funcions descrites a Processa(), per exemple **CALCULA OUR**.

#### 5.5.7.9. Assignacions i operacions amb variables.

Els noms de les variables es formen amb el símbol **\$** seguit de combinacions arbitràries de lletres, números i el caràcter **'\_'**. Opcionalment es pot especificar el format amb el modificador **%m.n**, on **m** i **n** són sencers que especifiquen la longitud total màxima (**m**) i el nombre de decimals (**n**).

Totes les variables són flotants. S'avaluen quan s'executa l'ordre en què s'utilitzen, substituint-les per un cadena amb el seu valor numèric, si no s'especifiquen uns altres valors, en format **%m.n**.

- Declaració de variables: Declara les variables **\$VAR1**, **\$VAR2**, etc. i s'expressa de la forma: **DEFINE***C***\$VAR1***C***\$VAR2** . . . **\$VAR***N**C* .
- Assignacions:
  - ✓ Assignació simple: **\$VAR1***C***=***C***EXPRESSIO** .
  - ✓ Assignació amb addició: **\$VAR1***C***+=***C***EXPRESSIO** .
  - ✓ Assignació amb subtracció: **\$VAR1***C***-=***C***EXPRESSIO** .
  - ✓ Assignació amb multiplicació: **\$VAR1***C***\****C***EXPRESSIO** .
  - ✓ Assignació amb divisió: **\$VAR1***C***/=***C***EXPRESSIO** .
- Expressions: Una *EXPRESSIO* és qualsevol combinació de noms de variables i operadors i/o crides a funcions. Alguns exemples són:
  - ✓ **-5**
  - ✓ **\$var + (\$var < 0)**
  - ✓ **\$var + \$var < 0**
  - ✓ **\$interval \* 2 + \$var + 4\*3**
  - ✓ **\$interval / 2**
  - ✓ **(\$valor > \$max) && (\$valor < \$min)**
  - ✓ **PROC CALCULA OUR**

- Variabls predefinides: Aquestes s'han definit per a les variables i paràmetres de la DCU del Biostat MCD, per reemplaçar el llenguatge propi que té la comunicació de la DCU.

✓ *Process Values*

\$DCU_PV_TEMP	Temperatura
\$DCU_PV_JTEMP	Temperatura camisa
\$DCU_PV_STIRR	Velocitat d'agitació
\$DCU_PV_MOTCR	% motor
\$DCU_PV_pH	pH
\$DCU_PV_pO2	pO <sub>2</sub>
\$DCU_PV_ACID	Consum d'àcid
\$DCU_PV_BASE	Consum de base
\$DCU_PV_AFOAM	Consum d'antiescumejant
\$DCU_PV_SUBS	Bomba FE-211
\$DCU_PV_AIRFL	Cabal d'aire
\$DCU_PV_REDOX	Potencial redox
\$DCU_PV_GAS1	Gas 1 (aire/O <sub>2</sub> )
\$DCU_PV_GAS2	Gas 2 (N <sub>2</sub> )
\$DCU_PV_pCO2	pCO <sub>2</sub>
\$DCU_PV_HARV	Volum recollit de biomassa
\$DCU_PV_FEEDP	Volum bomba d'addició

✓ *Controller Set-point/Mode*

\$DCU_Cx_TEMP	Temperatura
\$DCU_Cx_STIRR	Velocitat d'agitació
\$DCU_Cx_pH	pH
\$DCU_Cx_pO2	pO <sub>2</sub>
\$DCU_Cx_pO2GF	pO <sub>2</sub> Gas-Flow
\$DCU_Cx_FOAM	Antiescumejant
\$DCU_Cx_SUBS	Velocitat bomba substrat
\$DCU_Cx_JTEMP	Temperatura camisa
\$DCU_Cx_GASFL	Gas flow control
\$DCU_Cx_LEVEL	Nivell de líquid
\$DCU_Cx_AIRFL	Cabala d'aire
\$DCU_Cx_pO2GM	pO <sub>2</sub> Gas-Mix

x representa a la lletra 'S' per als punts de consigna i la lletra 'M' per als modes.

✓ *Digitals Inputs*

\$DCU_DI_FOAM	Nivell d'escumes
\$DCU_DI_LEVLO	Nivell de líquid baix
\$DCU_DI_LEVHI	Nivell de líquid alt
\$DCU_DI_TEMPC	Alarma T cultiu
\$DCU_DI_JTMPC	Alarma T termòstat
\$DCU_DI_HEATC	Alarma T bescanviador
\$DCU_DI_FUSEC	Alarma fusible
\$DCU_DI_SUPON	Unitat principal encesa

- Jerarquia dels operadors: La prioritat en què el mòdul Protocol realitza una sèrie d'operacions està determinat per un conjunt de regles que s'encarreguen d'eliminar les possibles ambigüitats que es poguessin presentar. Dins de cada nivell, l'avaluació es fa d'esquerra a dreta. El conjunt d'operadors implementats i la seva precedència és:

1. Crides a funcions: **PROC** (assumeix que la resta de l'expressió és l'argument de la funció).
2. Parèntesi: (, ). Si hi ha varis primer té prioritat els de més endins.
3. Operadors multiplicatius: \*, /, %. Tots ells amb igual prioritat.
4. Operadors additius: +, -. Ambdós amb igual prioritat.

5. Operadors lògics:  $>$ ,  $<$ ,  $==$ ,  $<=$ ,  $>=$ ,  $!=$ ,  $\&\&$ ,  $||$ . Tots ells amb igual prioritats.

- Funcions intrínseques: En el mòdul Express hi ha definides una sèrie de funcions intrínseques que donen suport matemàtic a la sintaxi de Sequen. A la Taula 5.5 s'inclouen les funcions implementades que es poden cridar des de l'arxiu de protocol:

Sintaxi Sequen	Àlgebra	Descripció
<b>log()</b>	$\text{Ln}(x)$	Logaritme neperià
<b>exp()</b>	$e^x$	Exponencial de base $e = 2.7183$
<b>fabs()</b>	$ x $	Valor absolut
<b>sqrt()</b>	$x$	Arrel quadrada
<b>ceil()</b>	-	Arrodoniment a un sencer per sobre
<b>floor()</b>	-	Arrodoniment a un sencer per sota
<b>sin()</b>	$\text{Sin}(x)$	Sinus trigonomètric
<b>cos()</b>	$\text{Cos}(x)$	Cosinus trigonomètric
<b>tan()</b>	$\text{Tan}(x)$	Tangent trigonomètric
<b>asin()</b>	$\text{Arcsen}(x)$	Arc sinus
<b>acos()</b>	$\text{Arccos}(x)$	Arc cosinus
<b>atan()</b>	$\text{Arctan}(x)$	Arc tangent

Taula 5.5. Funcions intrínseques utilitzades a la sintaxi de Sequen.

#### 5.5.7.10. Anàlisi d'arxius de protocol: Exemples.

- Exemple #1: Arxiu de protocol d'inici, INICI.TXT.

Com ja s'ha descrit a l'apartat 5.5.2.1, l'execució de l'arxiu de protocol INICI.TXT permet establir les comunicacions amb els diferents equips. El significat de cada ordre es descriu a continuació:

#Inicialitza la comunicació amb els equips: DCU de Braun, turbidímetre d'Aquasant, bomba d'Ismatec i electrovàlvula de Sirai.

INI DCU

INI BIOMASSA

INI BOMBA 0

INI ELECTROVALVULA 0

#Dirigeix en aquest accés tots els fitxers de registre.

```
SET PATH C:\XAVIER\BATCH
```

#Etiqueta de final de l'arxiu de protocol.

```
END
```

▪ Exemple #2: Subrutina del càlcul del consum d'oxigen, OUR\_On.

Aquesta operació, que s'ha descrit a l'apartat 4.5.2, amb el corresponent diagrama de flux de la seqüència d'anàlisi, es descriu ara a nivell del corresponent arxiu de protocol.

#Inicia la realització	#Comentari.
#d'una OUR en DCU-17	#Comentari.
LABEL OUR_On	#Nom de la subrutina.
SET RATE BIOMASSA 1	#Canvi de la velocitat d'adquisició del turbidímetre.
INC FILE BIOMASSA	#Increment de l'índex del fitxer del turbidímetre.
\$DCU_CS_pO2 = 100.0	#Canvi de consigna del controlador pO2GM.
\$DCU_CM_pH = 0	#Desactiva del controlador pH.
DELAY \$SAFE	#Retard amb un temps de valor \$SAFE = 150 s.
DCU:DS:0:0:CS:12:0	#Canvi de consigna de la variable pO2GM amb format de llenguatge DCU.
DELAY \$T_DEPRESS	#Retard amb un temps de valor \$T_DEPRESS = 30 s.
SET RATE DCU 1	#Canvi de la velocitat d'adquisició de les variables DCU.
INC FILE DCU	#Increment de l'índex del fitxer de la DCU.
ELECTROVALVULA 0:ON:1	#S'obre l'electrovàlvula 0.
WHILE \$DCU_PV_pO2 > \$LIMINF	#Itera fins que troba que el valor del pO2 és més baix que el valor definit a \$LIMINF = 29.5%, esperant entre cada comparació un temps \$T_COMP = 5 s.
DELAY \$T_COMP	
END WHILE	

```
ELECTROVALVULA 0:ON:0          #Es tanca l'electrovàlvula 0.
DCU:DS:0:0:CS:12:600:CM:3:1    #Canvi de consigna de pO2GM = 60% i el
                                controlador de pH activat amb format de
                                llenguatge DCU.
SET RATE DCU 20                 #Canvi de la velocitat d'adquisició de les
                                variables DCU.
SET RATE BIOMASSA 5             #Canvi de la velocitat d'adquisició del
                                turbidímetre.
PROC CALCULA OUR                #Càlcul del consum d'oxigen (OUR) a
                                partir de les dades de pO2 entre el 50 i el
                                30%.
PROC CALCULA BIO                #Estimació de la biomassa a partir de les
                                dades del turbidímetre.
RETURN                           #Retorna a l'arxiu d'on ha estat cridada la
                                subrutina OUR_On.
```

▪ Exemple #3: Arxiu de protocol principal, BATCH.TXT.

A l'apartat 6.2.1 es presenta un diagrama de flux de la seqüència d'ordres per a l'operació d'un cultiu en discontinu, que ara es descriu a nivell d'arxiu de protocol:

```
#Consigna els valors i mode dels controladors de la DCU amb el llenguatge de l'equip.
DCU:DS:0:0:CS:1:246.667:2:30:3:502:11:150:12:600
DCU:DS:0:0:CM:1:1:2:1:3:1:6:0:11:1:12:1
```

```
#Fixa el directori de registre, els noms del fitxers de sortida i la velocitat d'adquisició de la
biomassa i de les dades de la DCU.
```

```
SET PATH C:\XAVIER\BATCH
SET RATE DCU -1
SET FILE DCU DCU000.DAT
SET RATE BIOMASSA -1
SET FILE BIOMASSA BIO000.DAT
```

```
#Defineix i fixa els valors de tot el conjunt de variables necessàries per a la neteja del
sensor turbidimètric, per al càlcul de la OUR i en la iteració del procés .
```

```
DEFINE $N_REPEAT_1, $N_REPEAT_2, $AUGMENTA_TEMPS;
$N_REPEAT_1 = 300;
$N_REPEAT_2 = 40;
$AUGMENTA_TEMPS = 7200;
```

```
DEFINE $T_ESPERA, $T_LOAD, $T_PURGA, $BIO_DELAY;  
$T_ESPERA = 3.2;  
$T_LOAD = 69;  
$T_PURGA = 2;  
$BIO_DELAY = 17.8;
```

```
DEFINE $LIMINF, $LIMCAL, $LIMSUP;  
$LIMINF = 29.5;  
$LIMCAL = 30.0;  
$LIMSUP = 50.0;
```

```
DEFINE $T_DEPRESS, $T_COMP, $KDES, $SAFE;  
$T_DEPRESS = 30;  
$T_COMP = 5;  
$SAFE = 150;  
$KDES = 0.000130;
```

#Canvia els noms del fitxers de sortida en un increment de l'índex i la velocitat d'adquisició de la biomassa i de les dades de la DCU.

```
INC FILE DCU  
SET RATE DCU 20  
INC FILE BIOMASSA  
SET RATE BIOMASSA 5
```

#Fixa el valor del variable temps en el valor 000:00:00.

```
SET TIME 0
```

#Rull iteratiu 1 fins que s'assoleix el valor de la variable \$N\_REPEAT\_1.

```
REP $N_REPEAT_1
```

```
INC TIME $AUGMENTA_TEMPS
```

#Cada cop que passa per ací augmenta la variable temps un el valor de 002:00:00.

```
ELECTROVALVULA 1:ON:1
```

#S'obre l'electrovàlvula 1.

```
DELAY $T_PURGA
```

#Retard amb un temps de valor \$T\_PURGA = 2 s.

```
ELECTROVALVULA 1:ON:0
```

#Es tanca l'electrovàlvula 1.

```
#Rull iteratiu 2 fins que s'assoleix el valor de la variable $N_REPEAT_2.
REP $N_REPEAT_2
GOSUB FILE BUBFREE.TXT           #Crida a l'arxiu de protocol
                                  BUBFREE.TXT.
END REP                          #Etiqueta de final d'iteració del rull 2.
GOSUB OUR_On                     #Crida a la subrutina OUR_On.
WAIT TIME                        #Espera que la variable temps arribi com a
                                  mínim al valor 002:00:00.
END REP                          #Etiqueta de final d'iteració del rull 1.
```

```
#Desactiva el mode dels controladors de la DCU amb el llenguatge propi de l'equip.
DCU:DS:0:0:CM:1:0:2:0:3:0:6:0:11:1:12:0
```

```
#Etiqueta de final de l'arxiu de protocol principal.
END
```

```
#Conjunt d'ordres de la subrutina OUR_On que es descriu a l'Exemple #2.
LABEL OUR_On
#ORDRES
RETURN
```

- Exemple #4: Arxiu de protocol de retroneteja (*backflush*) del mòdul CellFlo: BACKFLUSH.TXT.

Aquest exemple s'ha extret de l'arxiu de protocol per a l'operació en continu amb perfusió (apartat 6.4).

```
#Condicional de l'alarma de nivell de líquid al bioreactor, en què si les variables
$DCU_DI_LEVHI i $DCU_DI_FOAM, senyals proporcionats pel sensors de nivell #1 i de
nivell #2 respectivament, són certes, la bomba FE-211 d'addició s'atura i, si són falses,
s'engega.
IF ($DCU_DI_LEVHI && $DCU_DI_FOAM)
    DCU:DS:0:0:CM:7:0
ELSE
    DCU:DS:0:0:CM:7:1
END IF
```

```
#Suma a la variable $BACK_count el valor de 1.
$BACK_count += 1;
```



#Condicional en què si les variables \$BACK\_COUNT és igual a 8, es processen les ordres definides dins seu. Per tant, això no passa fins que aquest arxiu no es crida 8 cops des de l'arxiu principal.

```
IF($BACK_count == 8)
```

```
    #S'engega la bomba 0 en sentit horari i a 50 de velocitat.
```

```
    BOMBA 0:SENTIT:1
```

```
    BOMBA 0:SPEED:50
```

```
    BOMBA 0:STOP:0
```

```
    DCU:DS:0:0:CM:6:0
```

```
    #Atura la bomba de sortida de la perfusió  
    (FOAM).
```

```
    DELAY $T_BACKFLUSH
```

```
    #Retard amb un temps de valor  
    $T_BACKFLUSH = 120 s.
```

```
    BOMBA 0:STOP:1
```

```
    #S'atura la bomba 0.
```

```
    DCU:DS:0:0:CM:6:1
```

```
    #Engega la bomba de sortida de la  
    perfusió (FOAM).
```

```
    $VS = -($T_BACKFLUSH/6)
```

```
    #Calcula el volum addicionat de medi.
```

```
    $BACK_count = 0;
```

```
    #Assigna a $BACK_count el valor de 0.
```

```
END IF
```

```
END
```

```
    #Etiqueta de final d'arxiu.
```

## 5.6. L'APLICACIÓ SUPERVIS.

La gran quantitat de dades que es generen al llarg d'un procés de cultiu cel·lular crea la necessitat de poder visualitzar-les en forma de la seva evolució amb el temps. La interfície gràfica d'usuari de l'aplicació Sequen, Seqüenciació d'ordres, està pensada per a la gestió d'equips i l'adquisició dels diferents senyals remotament, visualitzant-se aquests en temps real (fins a 600 dades) i no en el seu conjunt total; i per poder realitzar, amb el processament de les mesures, una acció o accions de control sobre el procés de cultiu cel·lular. No seria gaire complicat definir dins l'aplicació uns panells que permetin representar gràficament aquelles dades sorgides del processament i el control del procés. No obstant, la conveniència de no engrandir més el programari de control i gestió, per evitar possibles fallades indesitjables del sistema, aconsella la definició i construcció d'una aplicació paral·lela a Sequen.prj, Supervis.prj, destinada exclusivament a aquesta finalitat de supervisió.

L'aplicació **Supervis**, que corre independentment de l'aplicació **Sequen**, mostra en format gràfic totes les dades sorgides de l'aplicació **Sequen** i, a més a més, permet introduir i representar les dades obtingudes fóra de línia dels anàlisis de mostres del bioreactor, així com, en cas de detectar errors en algun valor d'OUR, poder realitzar l'anàlisi manual del consum d'oxigen dissolt i, per tant, poder establir les possibles causes de l'error i corregir-les després.

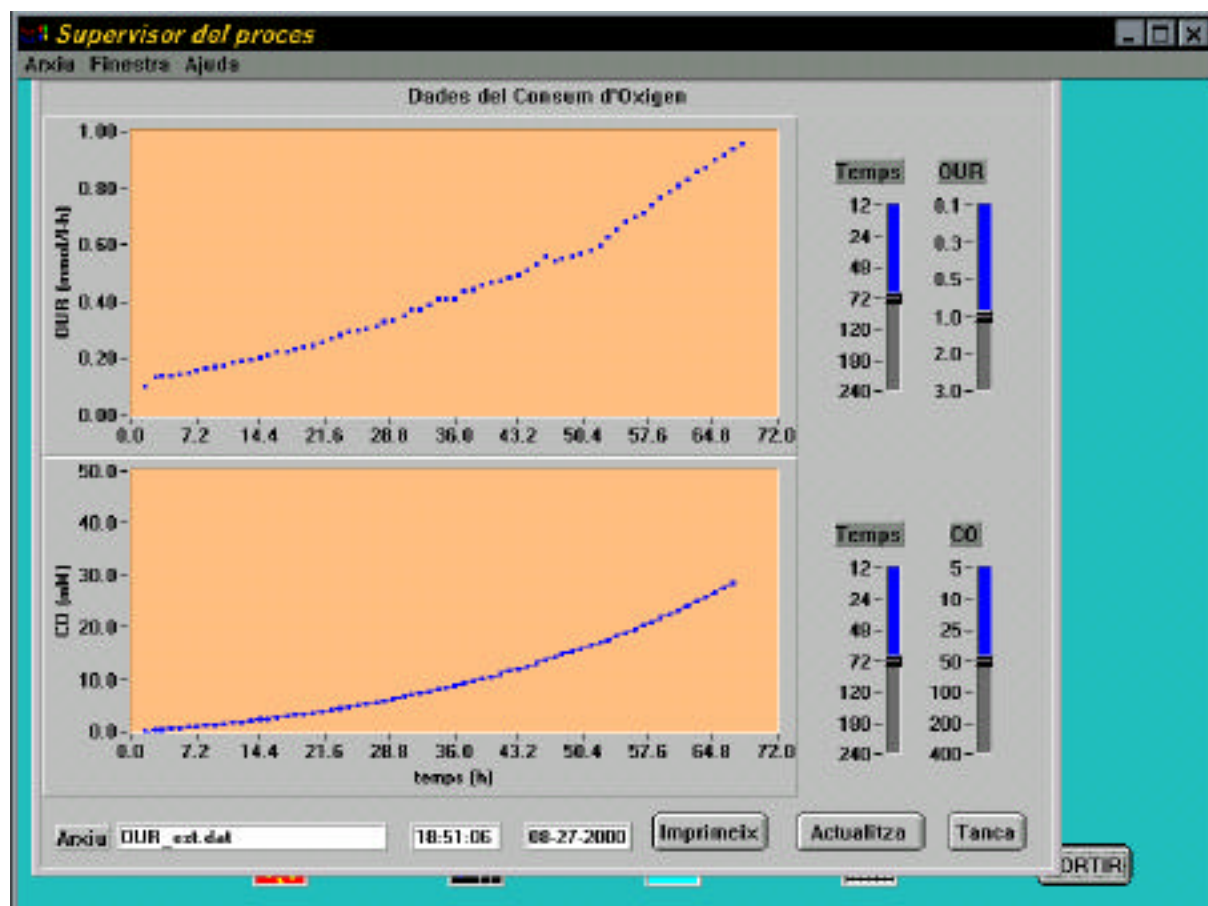


Figura 5.8. L'aplicació **Supervis** amb el panell obert de Dades del consum d'oxigen.

Com ja s'ha comentat a l'apartat 5.5.5, les dades estan registrades en fitxers de sortida amb el format adequat per a la seva lectura. Per tant, es programa un mòdul, anomenat **Supervis**, que és capaç de llegir i col·locar en matrius les dades de cadascun d'aquests fitxers, i representar-les en panells gràfics amb les funcions gràfiques de l'entorn **LabWindows/CVI**. A les figures 5.8, 5.9 i 5.10 es mostren els resultats finals del disseny de l'aplicació **Supervis** amb alguns dels seus panells oberts.

L'aplicació desenvolupada permet la interacció de l'usuari a partir d'una finestra única i uns panells que representen gràficament dins seu un determinat tipus de dades. Les dades es divideixen en cinc grups, i per tant en cinc panells: Dades del consum d'oxigen (OUR i consum acumulat d'oxigen (OC) respecte al temps) Dades del creixement cel·lular (Senyal del

turbidímetre i estimació de les cèl·lules totals respecte al temps), Dades de les mostres manuals (cèl·lules viables i totals, viabilitat, glucosa i lactat respecte al temps), Dades de la DCU (pH,  $pO_2$ , T,  $V_{NaOH}$ ,  $V_{CO_2}$  respecte al temps) i Anàlisi de la OUR ( $pO_2$  respecte al temps). La finestra Supervisor del procés de Supervis.uir, pot mostrar tots els panells alhora o un per un, a l'igual que es feia en Sequen. Les barres de selecció verticals permeten adequar el rang de mesura en funció de les necessitats de l'usuari, per a cadascuna de les variables. Sota els gràfics es mostra l'arxiu del qual procedeixen les dades, el dia i la hora. El botó Actualitza permet actualitzar manualment les noves dades del procés. Fent un simple clic sobre una de les icones gràfiques el panell s'obre i s'actualitza automàticament. També hi ha l'opció d'obrir i tancar panells amb la barra de menú Finestra. Els fitxers se seleccionen i carreguen automàticament, o també es pot obrir el menú de barres Arxiu i escollir-ne un del quadre diàleg de càrrega de fitxers.

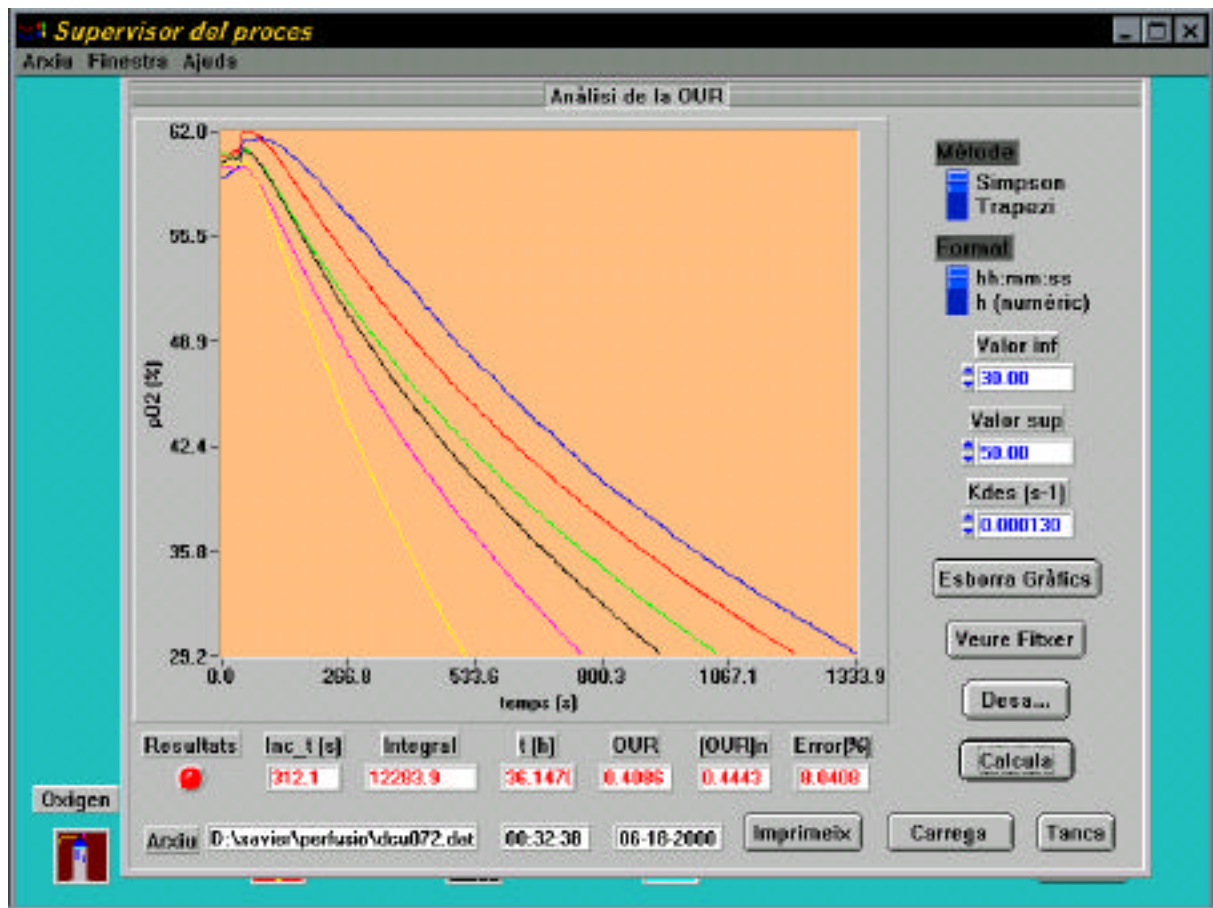


Figura 5.9. L'aplicació Supervis amb el panell obert de Anàlisi de la OUR.

En el panell Anàlisi de la OUR (Figura 5.9) es poden carregar amb el botó Carrega els diferents fitxers de dades adquirides de la DCU que es volen processar. En aquest panell existeixen caselles numèriques per fixar els paràmetres que es creguin oportuns, així com també interruptors binaris per escollir el mètode d'integració numèrica i/o el format de la

variable temps amb el que s'ha registrat el fitxer. Amb el botó **Calcula** es processa l'últim consum d'oxigen carregat en el panell, mentre que amb el botó **Desa...** es poden guardar els resultats obtinguts en un fitxer. Tots els resultats de l'últim fitxer analitzat es presenten en les caselles numèriques de sota el gràfic (LED encès quan els resultats són recents). Es pot visualitzar el fitxer de registre dels resultats desats (**Veure Fitxer**) i es pot esborrar el gràfic o gràfics que es mostren al panell (**Esborra Gràfics**).

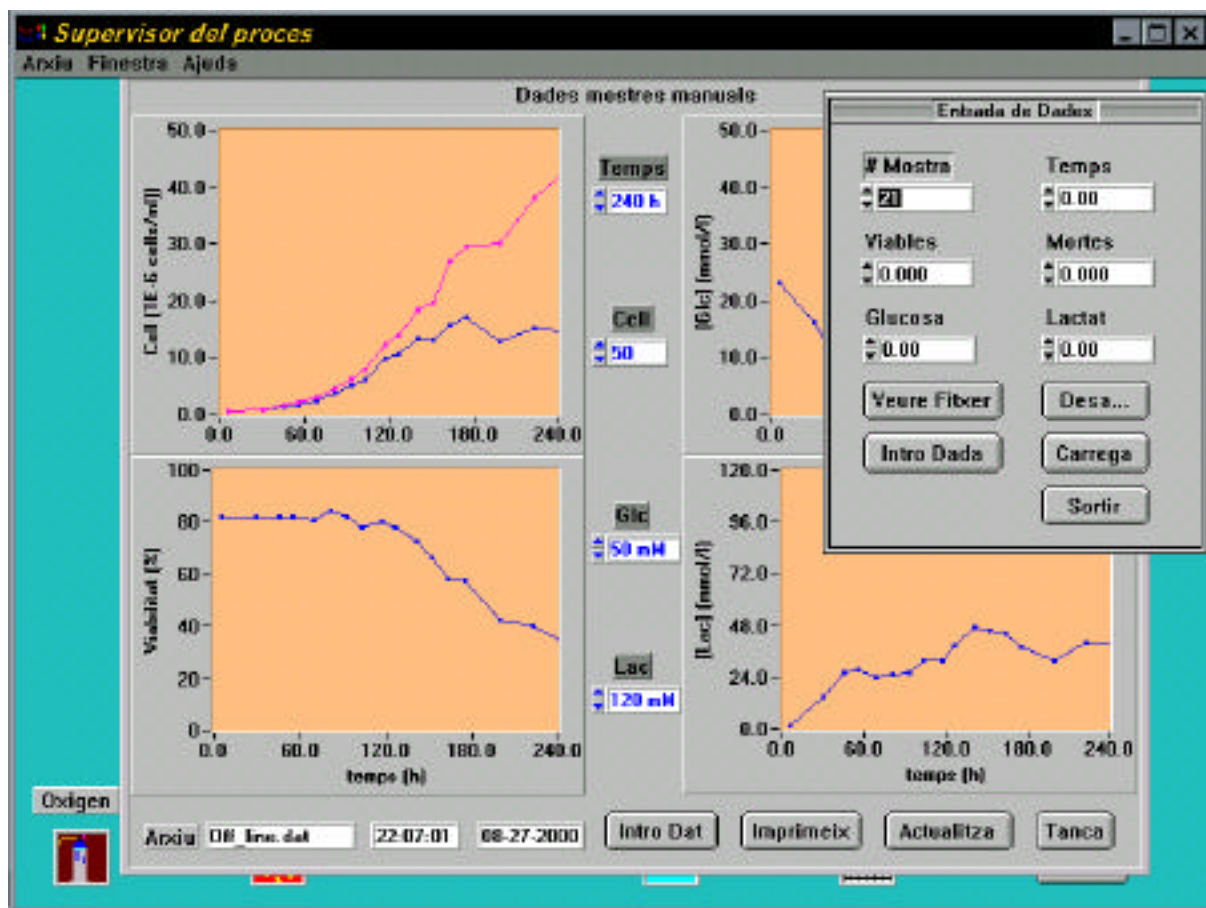


Figura 5.10. L'aplicació **Supervis** amb els panells oberts de **Dades mestres manuals** i **Entrada de Dades**.

El panell **Dades de mostres manuals** presenta el botó **Intro Dat** que permet visualitzar el panell **Entrada de Dades** (Figura 5.10). En aquest panell s'introdueixen els diferents valors de les anàlisis de mostres manuals que s'obtenen durant el cultiu cel·lular. Cada conjunt de dades que s'introdueix no es mostra fins que se selecciona el botó **Intro Dada**. El panell **Entrada de Dades** permet registrar les dades introduïdes en un fitxer de sortida (**Desa...**), així com visualitzar aquest (**Veure Fitxer**). També existeix la possibilitat de carregar un fitxer antic de dades amb el botó **Carrega**.

L'aplicació **Supervis** ha requerit la programació de 3 arxius de codi en llenguatge C i LabWindows/CVI (arxius \*.c + \*.h) amb prop de 2,000 línies que ocupen 63 kB de memòria.

Una vegada s'han desenvolupat i comprovat les eines informàtiques per poder gestionar i controlar el procés de cultiu cel·lular, s'està en disposició de treballar amb l'aplicació de les diferents estratègies de cultiu i addició de nutrients, en funció de les mesures disponibles en línia de la concentració i l'activitat de les cèl·lules d'hibridoma, amb la finalitat de definir quin tipus d'operació és la millor, és a dir, quina estratègia permet millorar la llargada dels cultius i, com a conseqüència, proporciona un major rendiment en la producció d'anticossos monoclonals.