References

[1] Gottfreid, B. S., **Elements of Stochastic Process Simulation**, Prentice-Hall, 1984, p. 8.

[2] Gordon, G., **System Simulation**, Prentice-Hall, 2nd edition, 1978, pp. 3-10.

[3] Freund, J.E., & Miller, I., **Probability and Statistics for Engineers**, Prentice-Hall, 3rd edition, 1985, pp. 178-180,222 & 284-286.

[4] Broctor, S.A., **Electric Circuit Analysis**, Prentice-Hall, 1987, pp. 532-555.

[5] Lynch, A.J., **Mineral Crushing and Grinding Circuits**, vol.1, chapter 2, Elsevier Publishers, 1977, pp. 21.

[6] Kleijnen, J.P., **Statistical Techniques in Simulation Part I**, chapter 1, Elsevier Publishers, 1974, pp. 6-12.

[7] Karra, V.K., **Development of a Model for Predicting the Screening Performance of a Vibrating Screen**, CIM Bulletin, April 1979, pp. 167-171.

[8] Canale, R. P., & Chapra, S. C., **Numerical Methods for Engineers with Personal Computer Applications**, McGraw-Hill, 2nd edition, 1987, p. 138.

[9] King, R.P., **Simulation - the modern cost effective way to solve crusher circuit processing problems**, International Journal of Mineral Processing, 1990, p.253-256.

[10] Pegden, C.D., Shannon, R.E., & Sadowski, R.P., **Introduction to Simulation Using Siman**, McGraw-Hill, 1990, p. 156.

[11] Georges, H., **Study of modelling and simulation for a chemical production system**, Simulation, June 1992, p.367-374.

Appendix A

Siman Model File

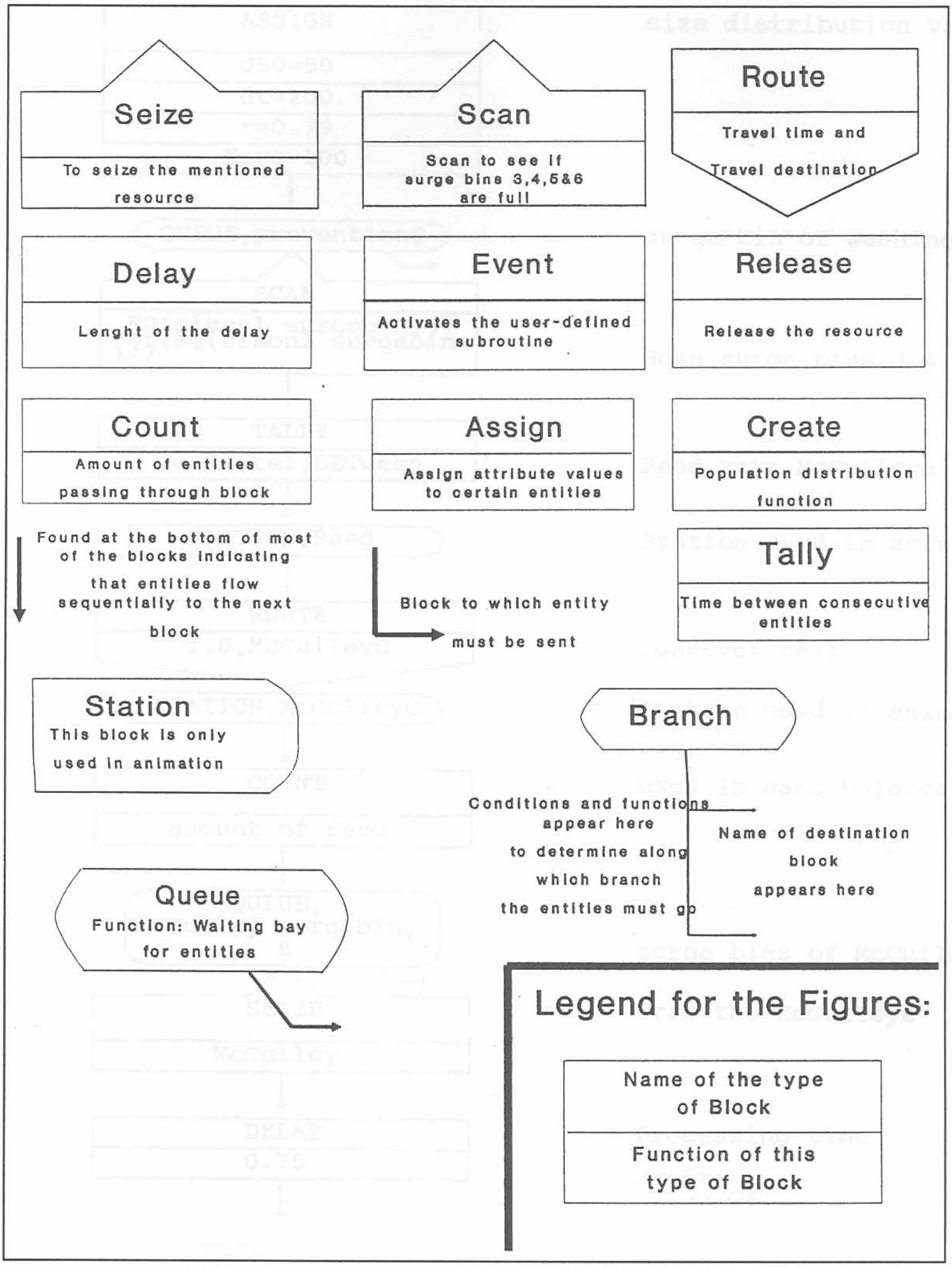
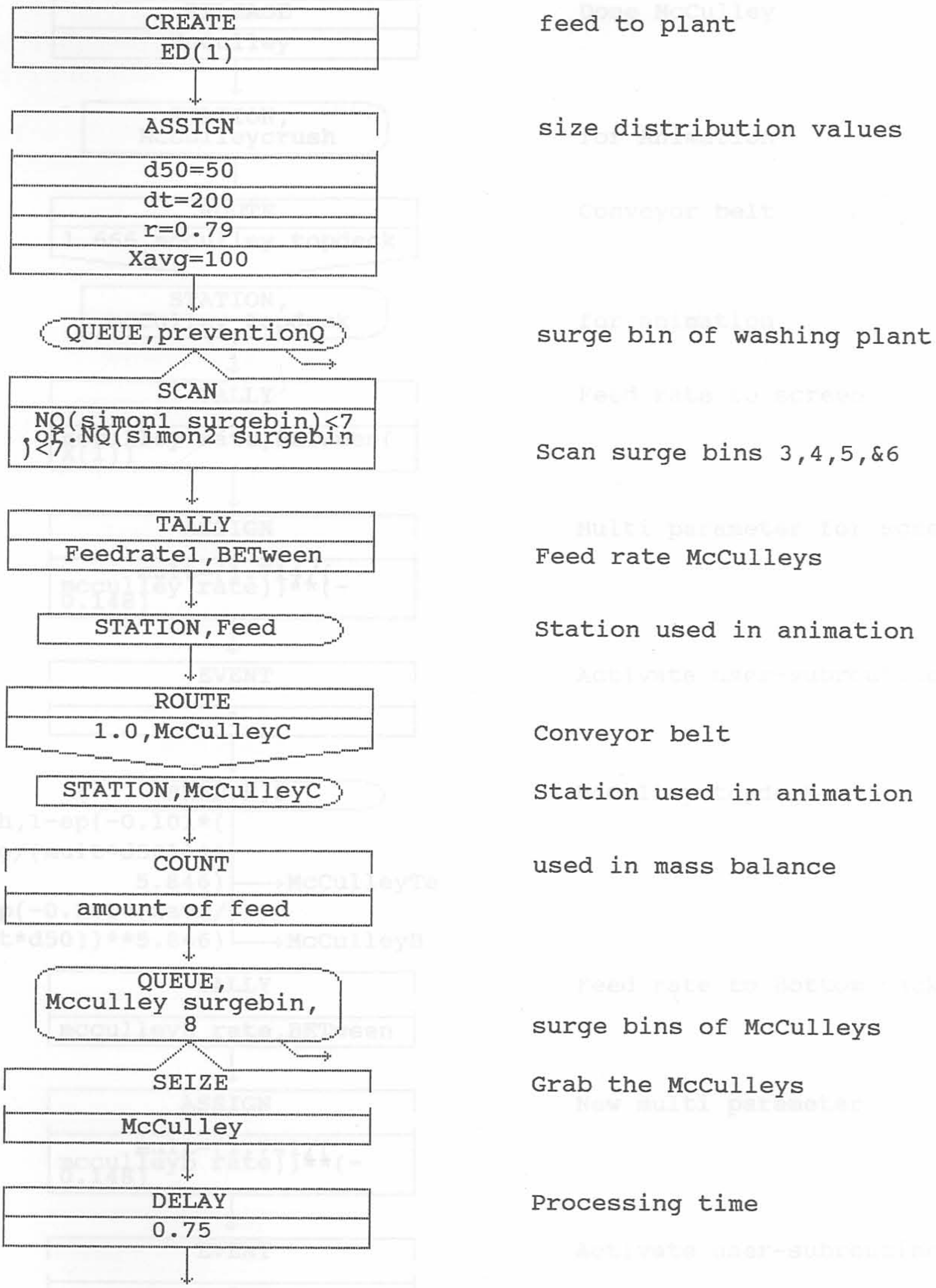
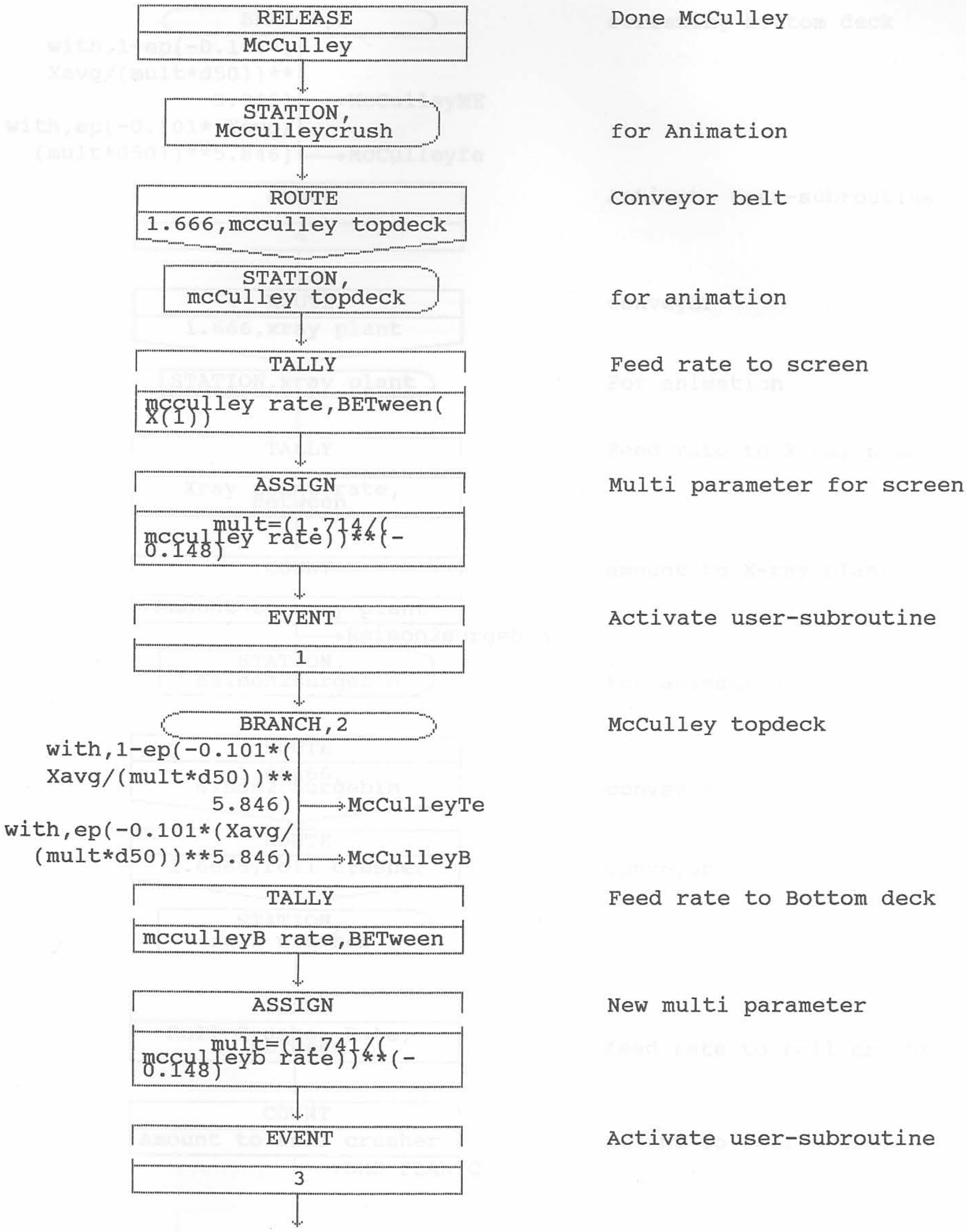
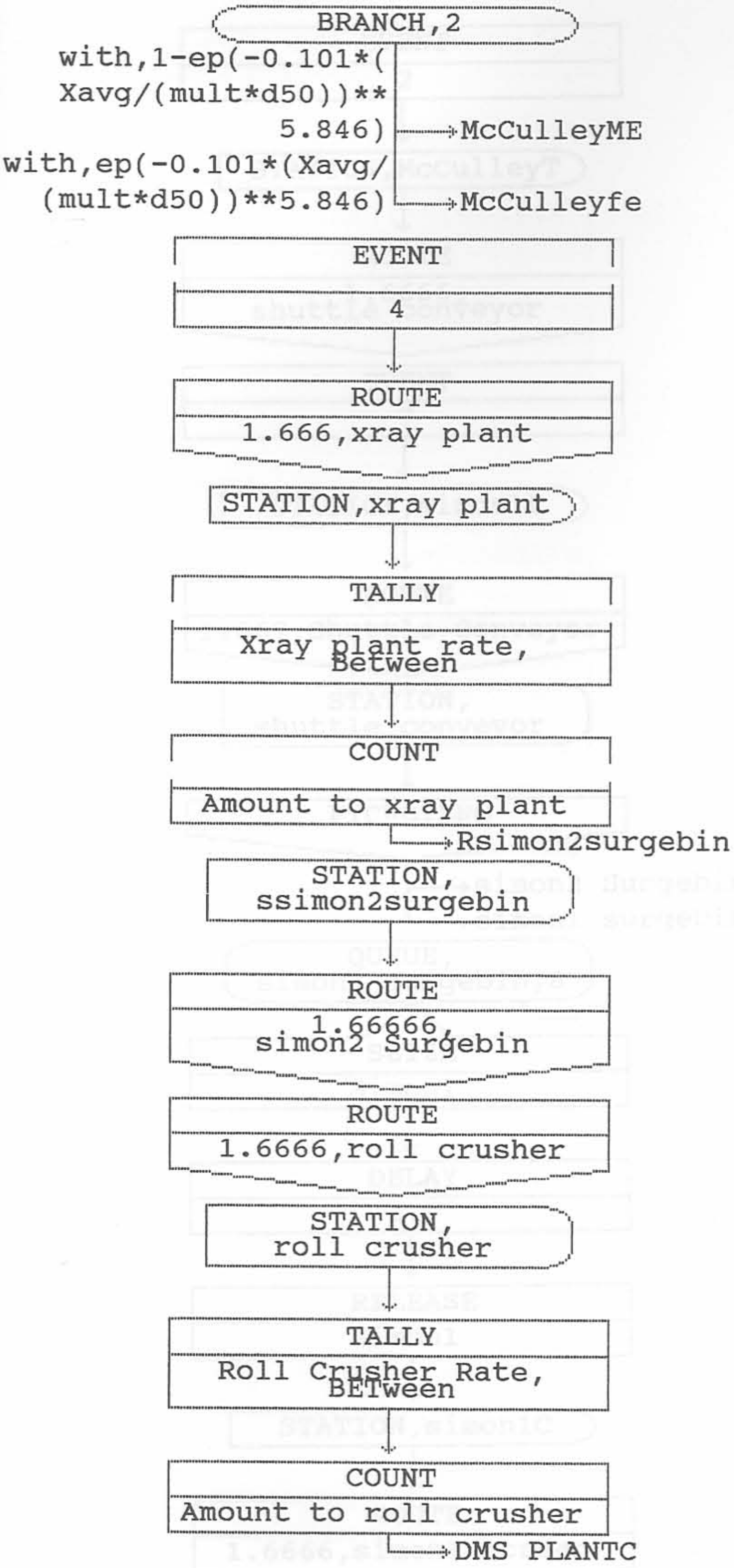

Seize
To seize the mentioned resource
Scan
Scan to see if surge bins 3,4,5&6 are full
Route
Travel time and Travel destination
Delay
Lenght of the delay
Event
Activates the user-defined subroutine
Release
Release the resource
Count
Amount of entities passing through block
Assign
Assign attribute values to certain entities
Create
Population distribution function
Found at the bottom of most of the blocks indicating that entities flow sequentially to the next block
Block to which entity must be sent
Tally
Time between consecutive entities
Station
This block is only used in animation
Branch
Conditions and functions appear here to determine along which branch the entities must go
Name of destination block appears here
Queue
Function: Waiting bay for entities
Legend for the Figures:
Name of the type of Block
Function of this type of Block

Figure A.1.- The list of blocks used and their functions in the Siman model file.


CREATE
ED(1)
feed to plant
ASSIGN
d50=50
dt=200
r=0.79
Xavg=100
size distribution values
QUEUE,preventionQ
surge bin of washing plant
SCAN
NQ(simon1 surgebin)<7 .or.NQ(simon2 surgebin)<7
Scan surge bins 3,4,5,&6
TALLY
Feedrate1,BETween
Feed rate McCulleys
STATION,Feed
Station used in animation
ROUTE
1.0,McCulleyC
Conveyor belt
STATION,McCulleyC
Station used in animation
COUNT
amount of feed
used in mass balance
QUEUE,
Mcculley surgebin,
8
surge bins of McCulleys
SEIZE
McCulley
Grab the McCulleys
DELAY
0.75
Processing time


RELEASE
McCulley
Done McCulley
STATION,
Mcculleycrush
for Animation
ROUTE
1.666,mcculley topdeck
Conveyor belt
STATION,
mcCulley topdeck
for animation
TALLY
mcculley rate,BETween(
X(1))
Feed rate to screen
ASSIGN
mult=(1.714/(
mcculley rate))**(-
0.148)
Multi parameter for screen
EVENT
1
Activate user-subroutine
BRANCH,2
McCulley topdeck
with,1-ep(-0.101*(
Xavg/(mult*d50))**
5.846)
McCulleyTe
with,ep(-0.101*(Xavg/
(mult*d50))**5.846)
McCulleyB
TALLY
mcculleyB rate,BETween
Feed rate to Bottom deck
ASSIGN
mult=(1.741/(
mccuIleyb rate))**(-
0.148)
New multi parameter
EVENT
3
Activate user-subroutine



screening bottom deck

Activate user-subroutine

conveyor

For animation

Feed rate to X-ray plant

amount to X-ray plant

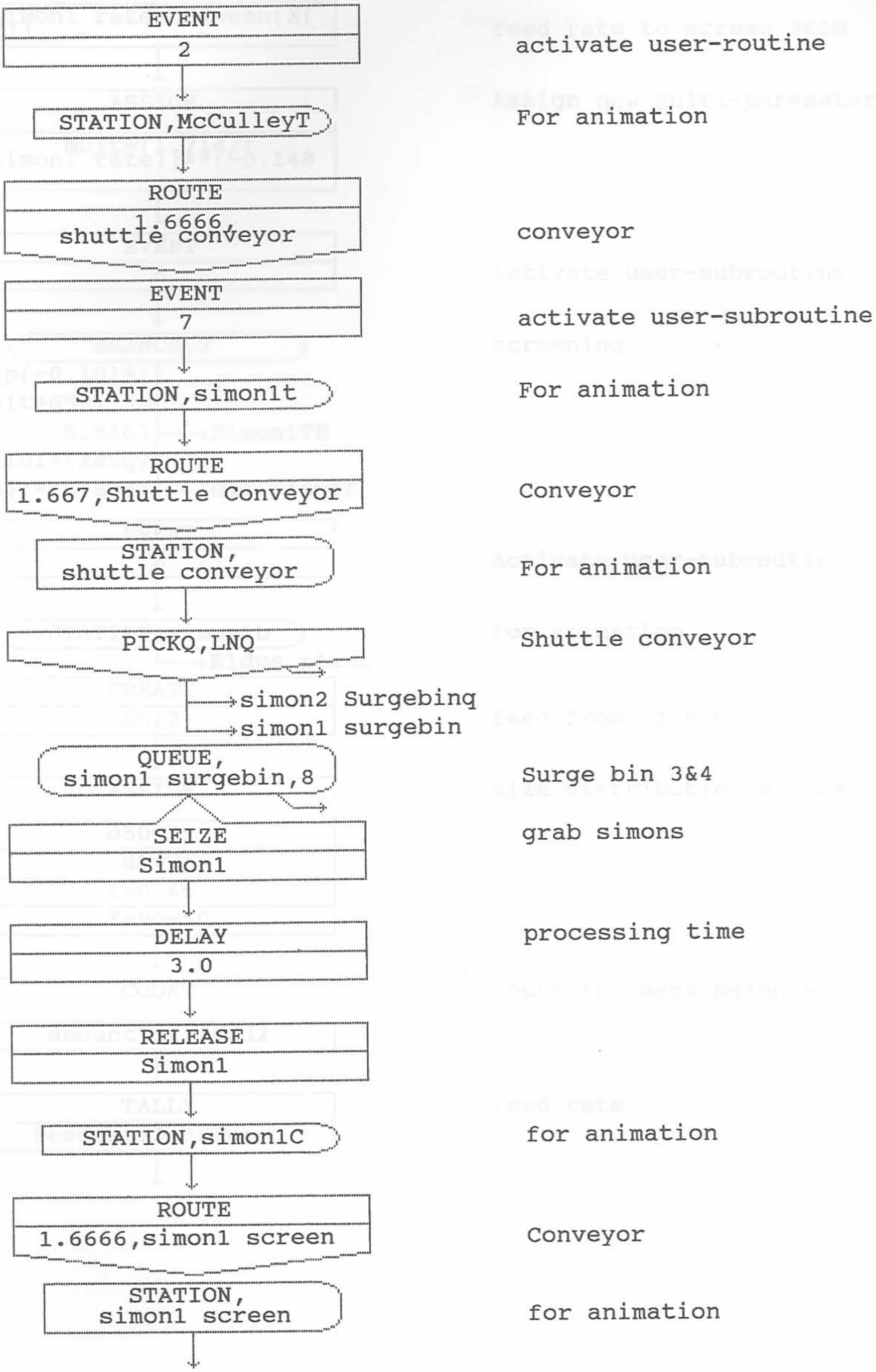
For animation

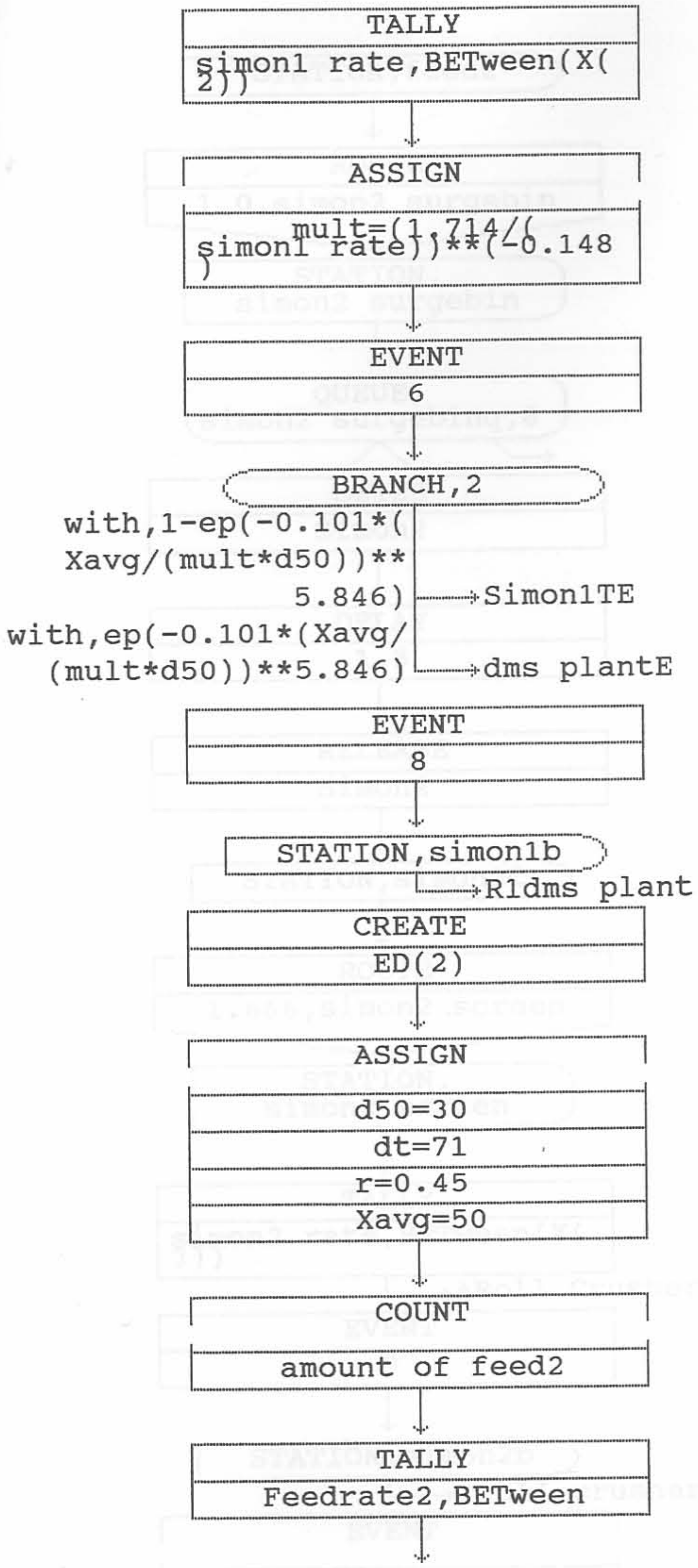
conveyor

conveyor

feed rate to roll crusher

amount to roll crusher


EVENT
2
activate user-routine
STATION,McCulleyT
For animation
ROUTE
1.6666,
shuttle conveyor
conveyor
EVENT
7
activate user-subroutine
STATION,simon1t
For animation
ROUTE
1.667,Shuttle Conveyor
Conveyor
STATION,
shuttle conveyor
For animation
PICKQ,LNQ
Shuttle conveyor
simon2 Surgebinq
simon1 surgebin
QUEUE,
simon1 surgebin,8
Surge bin 3&4
SEIZE
Simon1
grab simons
DELAY
3.0
processing time
RELEASE
Simon1
STATION,simon1C
for animation
ROUTE
1.6666,simon1 screen
Conveyor
STATION,
simon1 screen
for animation



feed rate to screen 9&10

Assign new multi-parameter

Activate user-subroutine

screening

Activate user-subroutine

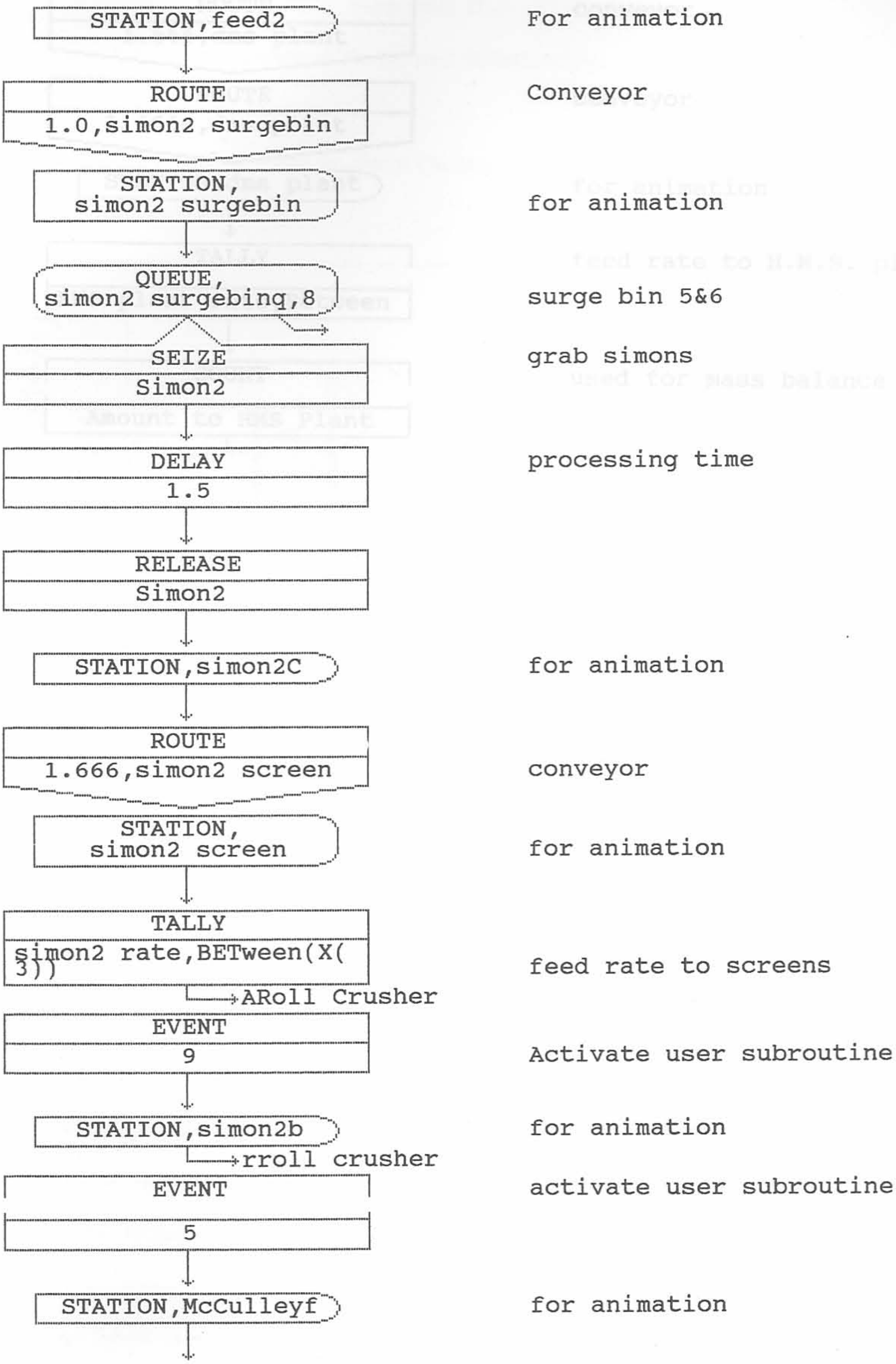
for animation

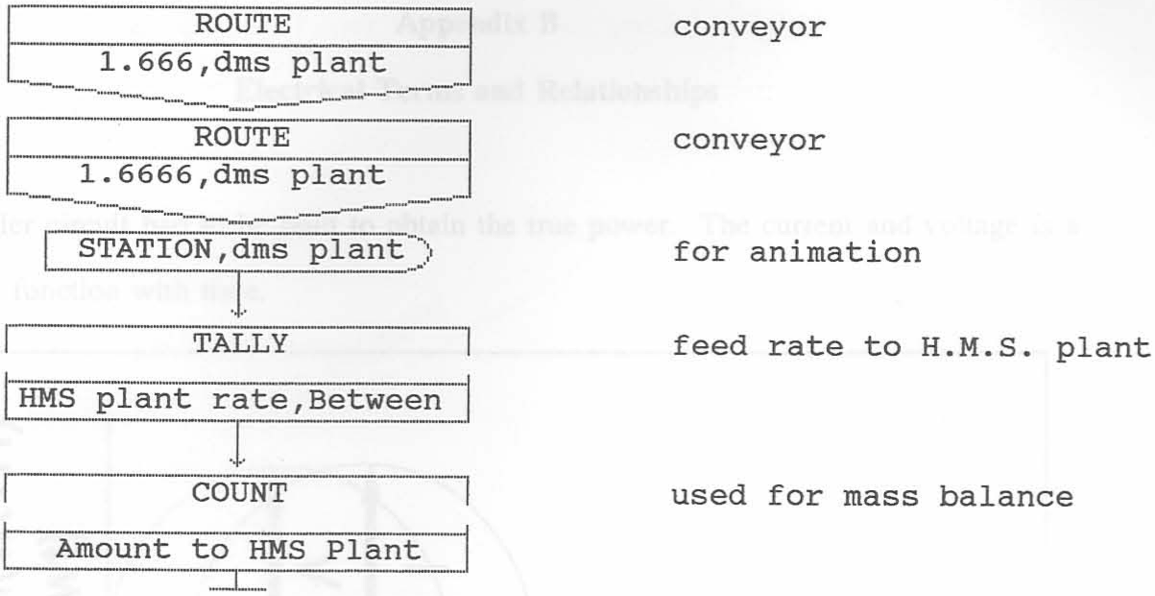
feed from wash plant

size distribution values

count for mass balance

feed rate


STATION,feed2
For animation
ROUTE
1.0,simon2 surgebin
Conveyor
STATION,
simon2 surgebin
for animation
QUEUE,
simon2 surgebinq,8
surge bin 5&6
SEIZE
Simon2
grab simons
DELAY
1.5
processing time
RELEASE
Simon2
STATION,simon2C
for animation
ROUTE
1.666,simon2 screen
conveyor
STATION,
simon2 screen
for animation
TALLY
simon2 rate,BETween(X(
3))
feed rate to screens
ARoll Crusher
EVENT
9
Activate user subroutine
STATION,simon2b
for animation
rroll crusher
EVENT
5
activate user subroutine
STATION,McCulleyf
for animation


ROUTE
1.666,dms plant
conveyor
ROUTE
1.6666,dms plant
conveyor
STATION,dms plant
for animation
TALLY
HMS plant rate,Between
feed rate to H.M.S. plant
COUNT
Amount to HMS Plant
used for mass balance

Appendix B

Electrical Terms and Relationships

A multiplier circuit had to be built to obtain the true power. The current and voltage is a sinusoidal function with time.

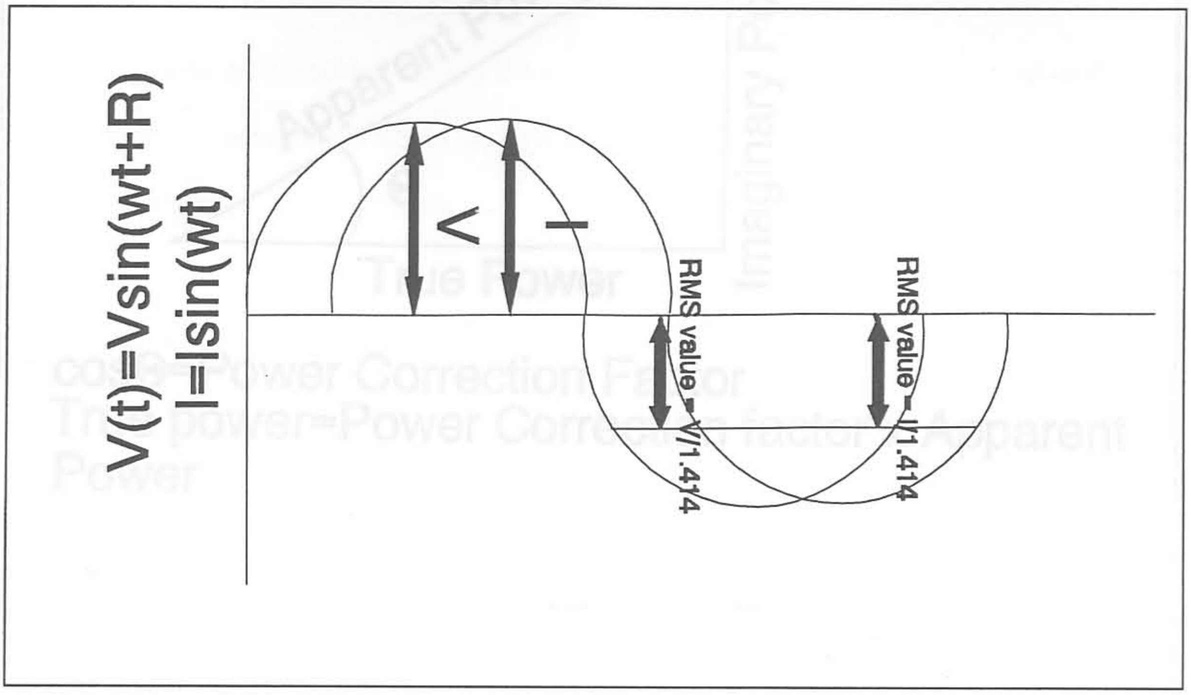


Figure B.1 - An example of the variation of the current and potential with time.

The current and voltage are not necessarily in phase, due to capacitance and inductance in the electric circuit. The multiplying the average current (RMS value) with the average voltage (RMS value) does not give the true energy consumption, because of the phase difference between the current and the voltage. By multiplying the immediate current with the immediate voltage this problem is overcome and the true power is obtained. The recorder that was linked to the multiplier had a capacitance in it which averages the output of the multiplier, so that immediate variations in the output were not recorded.

The following figure is a graphical representation of the relationship between the Power

Correction Factor, the Apparent Power, the True Power and the Imaginary Power.

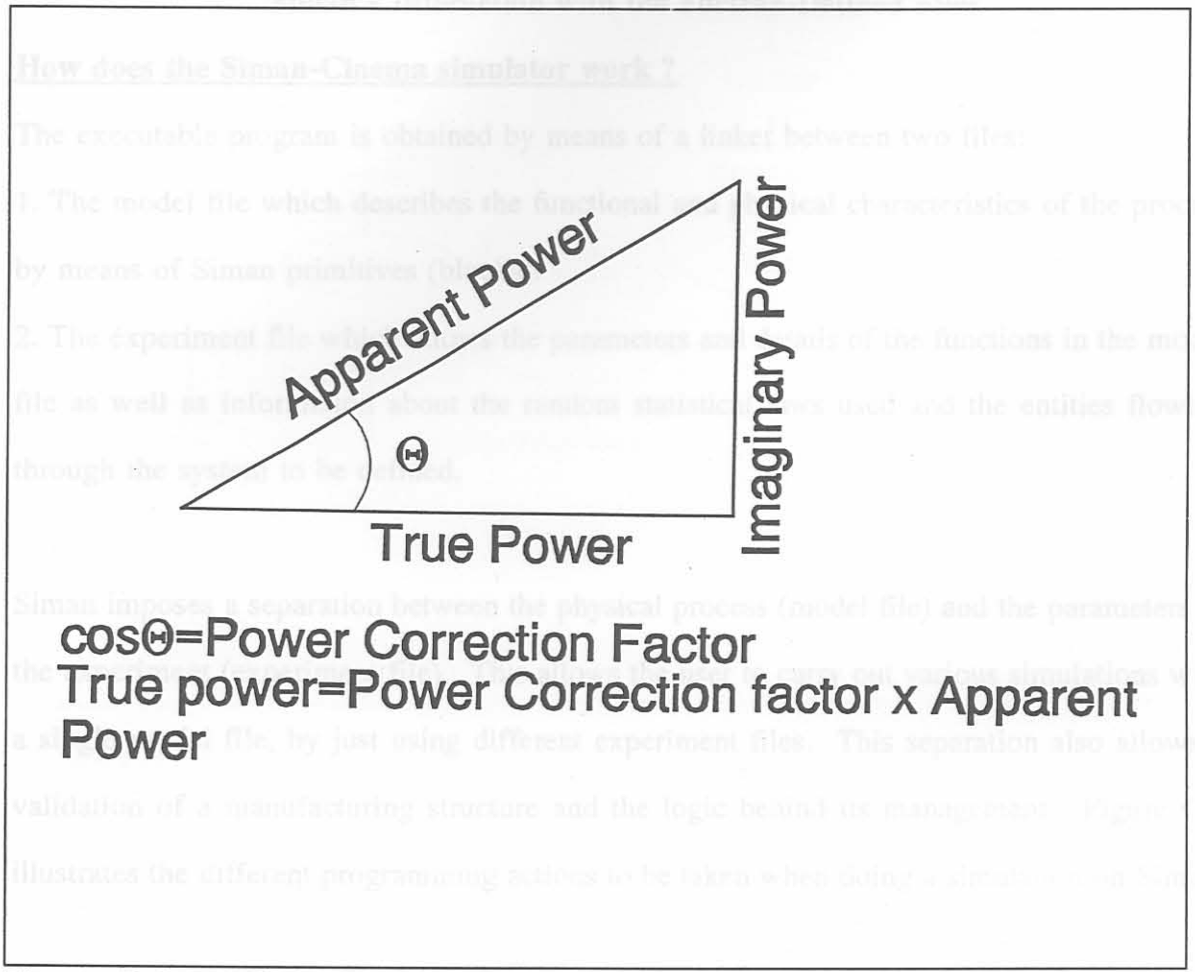

Apparent Power
Imaginary Power
Θ
True Power
cosΘ=Power Correction Factor
True power=Power Correction factor x Apparent Power

Figure B.2 - Relationship between the different definitions in Electrical Power Consumption.

Appendix C

Siman's Interaction with the Fortran-Defined Files

How does the Siman-Cinema simulator work ?

The executable program is obtained by means of a linker between two files:

1. The model file which describes the functional and physical characteristics of the process by means of Siman primitives (blocks).

2. The experiment file which allows the parameters and details of the functions in the model file as well as information about the random statistical laws used and the entities flowing through the system to be defined.

Siman imposes a separation between the physical process (model file) and the parameters of the experiment (experiment file). This allows the user to carry out various simulations with a single model file, by just using different experiment files. This separation also allows a validation of a manufacturing structure and the logic behind its management. Figure C.1 illustrates the different programming actions to be taken when doing a simulation on Siman.

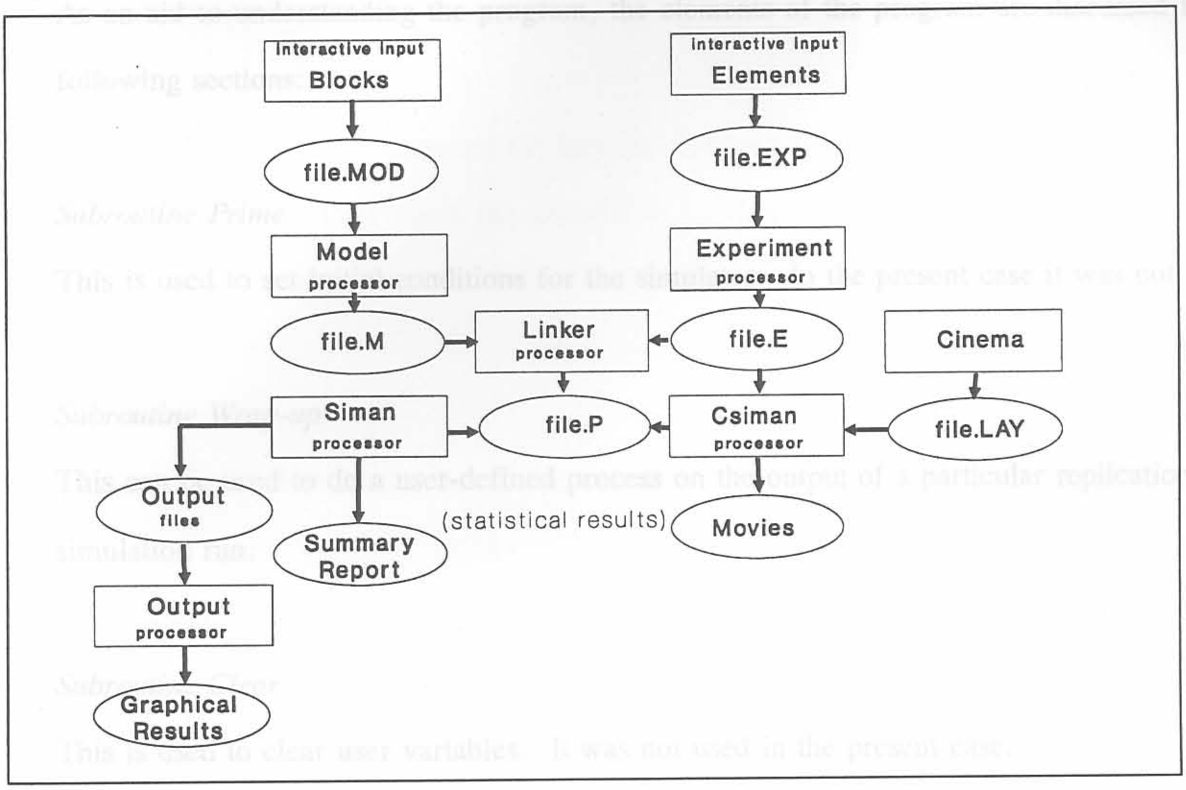


Figure C.1 - Siman-Cinema software structure.

How does the Simulation work in this specific study ?

Due to the complexities of simulating a metallurgical system a couple of unique subroutines were written in Fortran; this was written in such a manner as to create a new Siman executable file.

What happens is that as the simulation is running the siman simulation will be stopped if something predefined happens in the simulation. Then values will be passed onto the subroutines in the program that was written in Fortran until the subroutines in Fortran have completed their calculations and then the values are passed back to the Siman simulator. After which the simulation continues in Siman until another predefined action stops the simulation to send data to the program written in Fortran.

As an aid to understanding the program, the elements of the program are discussed in the following sections:

Subroutine Prime

This is used to set initial conditions for the simulator. In the present case it was not used.

Subroutine Wrap-up

This can be used to do a user-defined process on the output of a particular replication of a simulation run.

Subroutine Clear

This is used to clear user variables. It was not used in the present case.

Subroutine State

The subroutine State is used to calculate the energy consumption of the plant. The alpha, beta and lambda values are used to calculate the tonnage feed rate to particular units.

S(2), S(5) and S(8) calculate the energy consumption for the crushers using the empirical correlations stated in the main text.

S(3) and S(6) calculate the energy consumption of the screens using the empirical correlations stated in the main text.

S(10), S(11) and S(12) calculate the energy efficiency by dividing the energy consumed by the tonnage through the unit.

S(13) is the sum of the energy consumed of all the machines taken into consideration in this particular study.

S(14) is used to take into account the fact that for long times there is nothing flowing into the plant, while at other times the material is flowing into the plant as described by the population distribution function as described in the main text.

Subroutine Usav

Usav is called before a snapshot of user variables is taken.

Subroutine URST

Subroutine URST is called just after a snapshot to allow restoring status of the user variables.

Subroutine Keyhit

Keyhit is called up when a user-defined key is hit on the keyboard to run a user-defined function while the simulation is running.

Function UF(lent, nuf)

The function UF(lent, nuf) is used to set up a user-defined function in an expression in the model file of the Siman simulator.

Function UR(lent, nuf)

The function UR(lent, nur) is used to set up a user-defined rule. An example of this is the queue selection rule, i.e. rules that determine to which queue entities will be sent by the pickQ-block in the model file of the Siman simulator.

Subroutine Event

This is used to calculate the attributes of the entities used to describe the size distribution.

What happens while the simulation is running, is that as an entity runs through an event block in the Siman simulator the Event subroutine gets activated upon which two integer values are immediately passed on by the Siman simulator to the subroutine. These two integers are "lent" and "nev". Lent refers to the location of the calling entity (record pointer). Nev refers to the number of the event subroutine. The nev number is then used to determine which part of the subroutine must be run.

For example if an entity goes through the first event block the nev integer with the value of one it gets passed onto the subroutine. The statement "Goto (100,110,120,130,140,150,160,170),Nev" is used to run the part of the program marked with the 100.

Values of Attributes set due to crushing by McCulleys

Values get assigned to certain variables based on the value of the attributes of the entity, which entered the event block. The r-variable is assigned the value of the third attribute of the entity entering the event block by the function A(lent,3). This r-value is a measure of the inclination of the size distribution curve for the Gaudin-Meloy equation for the material of the entity. The dt-variable is assigned the value of the second attribute of the entity entering the event block by the function A(lent,2). This variable is the value for the top size of the material in the entity.

The variables n, n_1, n_2 and G are assigned the values which where determined using the Microsim simulator for the McCulley crusher.

The CCS variable refers the closed side setting for the McCulley crushers. The value of Xo is equal to the largest particle that will come out of the crusher, and this is related to the CCS. The values of d_1 and d_2 are related to the closed side setting and is used in the selection function.

The value of p is obtained using the breakage and selection as described in the main text. The value of p is equal to the value of r in the Gaudin-Meloy function for the product of the McCulley crushers. The log function is used because of the exponential nature of the r-value in the Gaudin-Meloy size distribution function.

The "call Seta(lent,3,p)" instruction sets the new value of the third attribute of the entity equal to the value of p.

The "call Seta(lent,2,Xo)" instruction sets the new value of the second attribute of the entity that entered the event block equal to the value of Xo.

Values calculated for screen to classify material

In this part of the program the values are calculated for those parameters which are used by the branch block to split the material.

The value of the aperture size of the screen is assigned to the Xap variable. The value of the inclination of the screen is assigned to the parameter Theta. The value of r is once again a measure of inclination of the size distribution curve. The value of "diam" is a measure of the wire diameter of the screen. The value of "U" is the material bulk density and this value was obtained using the Microsim simulator's optimization function. All these above mentioned values are used to calculate the d_{50} which is used in the branch block of the Siman simulator to split the material. The basic capacity factors, indicated by the parameters A_F,B_F,C_F,D_F,F_F and E_F in the program, are used to calculate the d_{50} which are obtained using the equations defined in the main text. The d_{50} is indicated by X50 in the program, the equation used to calculate X50, indicated in equation 17, correlates very closely with the equation indicated in the main text for d_{50}.

$$X50=ht*((305*(1-Q))/(A_F*B_F*C_F*D_F*E_F*F_F*8.33))**(-0.148) \qquad (23)$$

The major difference between these equations is that the "theoretical amount reporting to the under size per square meter" is calculated by "350*(1-Q)/8.33". Value of 350 is the standard tonnage approaching the screen, but the multi parameter converts this to the true amount approaching the screen at any given time using the tally block in the program for the model file indicated in appendix A.

The "call Seta(Lent,1,X50)" sets the value of the first attribute to the value of X50 so that this value can be used in the Siman simulator in the branch block (see appendix A) to determine the split of material.

Next follows an iteration routine using an If-statement to calculate the average size of the material in the entity. Because the Gaudin-Meloy function is a continuous function it had to be discretized to calculate the average size. As can be seen in this subroutine the average size is calculated by multiplying the percentage of material in each discretized size range by the average size of the size range and adding these values. The "call seta(lent,4,Sum)" sets the value of the fourth attribute to the value of sum and this value is used in the branch block of the Siman simulator as the average size to determine the split of the material.

The "Go To 1000" statement now sends the program to the end of the program and this action returns control to the Siman simulator to continue the simulation.

Values for top size from double deck screen

If an entity passes through the second event block in the model file of the Siman simulator the value of 2 gets passed on as the integer, Nev, to the Fortran program. The instruction "Go to (100,110,120,130,140,150,160,170,180), Nev" at the beginning of the program will now process the part marked with 110 due to the value of the Nev integer.

The Xo, X50 and r get assigned the values of the attributes of the entities using the function A(lent,number of the attribute).

The Xim-parameter is used to find the other point on the size distribution curve, described as (Xim,Yim), used to find the value of r for the Gaudin-Meloy equation as explained in the main text. The value of C is obtained using the classification function of the model of Karra as explained in the main text. Ywor is used to determine the fraction of material around Xim in the discretized size distribution function in the feed to the screen. Yim represents the fraction of material reporting to the oversize in the discretized size range, obtained by multiplying C and Ywor. This value is used to calculate the value of r in the Gaudin-Meloy size distribution function using the Newton-Raphson iteration technique.

Error is a value used to assure a good accuracy for the value of p calculated in the iteration routine. The iteration routine is done in the program using the If-statement. The value of p is assigned to the third attribute of the entity of the event block by the "call seta(lent,3,p)" instruction which is a measure of the inclination of the size distribution curve, the value of r. The Goto 1000 statement sends transfers control back to the Siman simulator and simulation continues.

Feed to bottom deck

If an entity passes through the third event block in the model file of the Siman simulator the value of 3 gets passed on as the integer, Nev, to the Fortran program. The instruction "Go to (100,110,120,130,140,150,160,170,180), Nev" at the beginning of the program will now process the part marked with 120 due to the value of the Nev integer.

The values X50, Xim, r and C are used and defined as mentioned above. The first value for Xo is the that of the attribute for entity that has entered the event block and is used to calculate Ywor, after this Xo is assigned the value of the top deck's aperture size. The value of Yim refers to the fraction of material reporting to the under size in a specific size range and is obtained by multiplying Ywor with (1-C). The value of p is worked out in the same manner described above. The values of the third and second attribute are set by the "call seta(lent,3,p)" and "call seta(lent,2,Xo)" instructions respectively.

The X50 is calculated in the same manner as described above using the basic capacity factors. The value for the average size of the size distribution curve is also calculated in the same manner as mentioned above. The "call seta" instruction sets both the values for the new attributes which are used in the branch block of the Siman model file as indicated in appendix A.

The "Go To 1000" instruction now goes to the end of the event subroutine and control is passed back to the Siman simulator and the simulation continuous.

The Middle size from the Double deck screen

This is calculated with the same technique as described above with the Newton-Raphson iteration technique used to calculate the value of the inclination of the size distribution curve.

Under Size from the Double Deck Screen

This is once again basically a repeat of the above mentioned calculations.

Values for due to Crushing of Symon 1 & 2 Crusher

This is calculated in the same manner as for the McCulley crushers. The values of some of the parameters have been changed based on the values found by the Microsim optimization function such as the values of G,n,n_1 and n_2.

Values set for the screen of Symon 1&2 to classify

This is also calculated in the same manner as for the material about to be classified by the double deck screen.

Top and Bottom Size from Screen of Symon 1&2 Crusher

This is just a repeat of the technique described for the double deck screen.

Value for Crushing by Symon 5&6 crusher

This is done in the same manner as for Symon 3&4 crusher.

```
c *******************************************************************
c
c Module :         USERFOR.FOR
c
c Description :    Dummy (stub) user FORTRAN routines
c
c Routines :       PRIME  - initial conditions/initial event setup
c                  WRAPUP - end of replication logic
c                  UCLEAR - clear user-defined statistics
c                  STATE  - state and differential equations (continuous)
c                  USAV   - save current status of the system
c                  URST   - reset saved status of the system
c                  KEYHIT - process key hit during simulation
c                  UF     - user function (called by UF siman variable)
c                  UR     - user rule (called by UR siman variable)
c                  EVENT  - event number mapped to appropriate event routine
c
c Global Variables:
c
c /sim/          General user-accessable common block
c
c       d        derivative value
c       dl       last derivative value
c       s        state variable value
c       sl       last state variable value
c       x        global variable array
c       dtnow    current integration step size
c       tnow     current simulated time
c       tfin     ending time of current run
c       j        integer global variable
c       numrep   current replication number
c
c     common/sim/d(50),dl(50),s(50),sl(50),x(50),
c    1     dtnow,tnow,tfin,j,numrep
c
c
c Note: IDUM assignments are included only to prevent compiler warnings
c
c Copyright 1989, Systems Modeling Corporation.  All Rights Reserved
c *******************************************************************
c
c
c     ---------------- ERROR DETECTION --------------------------------
c
c     We recommend uncommenting the following "include" statement and
c     placing a similar statement at the top of all FORTRAN files used
c     with SIMAN.  This makes use of FORTRAN extensions (not available
c     on all platforms, UNIX) to allow checking of the number and type of
c     arguments for all SIMAN user-callable routines.  Argument errors
c     are often undetected by either the compiler or user and may
c     cause unpredictable results.  Consistent use of simlib.f in
c     every file can reduce or eliminate this type of error.
c
c
c     include 'simlib.f'
c
c
c     -----------------------------------------------------------------
c
c
```

```
      subroutine PRIME
c     ***************************************************************
c
c     Called at the beginning of each replication to process user
c     initialization code.
c
c     ***************************************************************
      return
      end
c
c
      subroutine WRAPUP
c     ***************************************************************
c
c     Called at the end of each replication to process user logic
c     associated with the end of the replication.
c
c     ***************************************************************
      return
      end
c
c
      subroutine UCLEAR
c     ***************************************************************
c
c     Called by SIMAN’s CLEAR routine to also clear user variables.
c
c     ***************************************************************
      return
      end
c
c
      subroutine STATE
c     PRIMARY CRUSHING PLANT
c
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      Common/user/lamda,beta,alpha
c
      S(1) = X(1)
      S(4) = X(2)
      S(7) = X(3)
      If(S(1).GT.SL(1)) then
       Lamda = (S(1)-SL(1))
      Endif
      If(S(4).GT.SL(4)) then
       Beta = (S(4)-SL(4))
      Endif
      If(S(7).GT.SL(7)) then
       alpha = S(7)-SL(7)
      Endif
      If(lamda.GT.0) then
       S(2) = (17.80+61.7292/(2*lamda))*2
       S(3) = (3.30+2.3448/(2*lamda))*2
       S(10) = (S(2)+S(3))*lamda/600
      Endif
      If(beta.GT.0) then
       S(5) = (57.96+77.934/beta)
       S(6) = (3.30+2.3448/beta)
       S(11) = (S(5)+S(6))*beta/600
      Endif
      If(alpha.GT.0) then
       S(8) = (57.96+77.934/(2*alpha))*2
       S(9) = (3.30+2.3448/(2*alpha))*2
       S(12) = (S(8)+S(9))*alpha/600
      Endif
      S(13) = S(2)+S(3)+S(5)+S(6)+S(8)+S(9)
      S(14) = (cos(0.418879*Tnow))**2-0.0000055
      If(S(14).GT.0) then
       X(4) = 1
      Else
       X(4) = 100
      Endif
      return
      end
C
C
      subroutine USAV
C     ***************************************************************
C
C     Called just before a snapshot is saved to allow saving status
C     of user variables.
C
C     ***************************************************************
      return
      end
C
C
      subroutine URST
C     ***************************************************************
C
C     Called just after a snapshot is recalled to allow restoring
C     status of user variables.
C
C     ***************************************************************
      return
      end
C
C
      subroutine KEYHIT(icode)
C     ***************************************************************
C
C     Called before any user keystoke is interpreted by SIMAN.
C     Resetting icode to 0 inhibits additional processing of key press
C
C     icode  : >0 : ASCII Keys 1-128 (on some systems 1-255)
C
C              <0 : System specific keys (generally as follows):
C
C              -1 : Function Key 1      |  -15 : HOME Key
C               :       :               |  -16 : PGUP Key
C             -10 : Function Key 10     |  -17 : PGDN Key
C             -11 : Up Arrow Key        |  -18 : END Key
C             -12 : Down Arrow Key      |  -19 : INS Key
C             -13 : Right Arrow Key     |  -20 : DEL Key
C             -14 : Left Arrow Key      |  -21 : SHIFT TAB Key (Back tab)
C
C     For the VAX: <0 :
C              -1 : PF Key 1            |  -11 : Up Arrow Key
C              -2 : PF Key 2            |  -12 : Down Arrow Key
C              -3 : PF Key 3            |  -13 : Right Arrow Key
```

```
c         -4  : PF Key 4              |    -14  : Left Arrow Key
c
c     ******************************************************************
      idum=icode
      return
      end
c
c
      function UF(lent,nuf)
c     ******************************************************************
c
c     Called by referencing function UF(nuf) within a SIMAN expressior
c     to return a value calculated by the user.
c
c        lent  : location (record pointer) of the calling entity
c        nuf   : number of user function to be processed
c
c     ******************************************************************
      idum=lent
      idum=nuf
      UF=0
      return
      end
c
c
      function UR(lent,nur)
c     ******************************************************************
c
c     Called by referencing function UR(nur) within a SIMAN rule
c     to return a value calculated by the user.
c
c        lent  : location (record pointer) of calling entity
c        nur   : number of user rule to be processed
c
c     ******************************************************************
      idum=lent
      idum=nur
      UR=0
      return
      end
c
c
      subroutine EVENT(lent,nev)
      Integer lent,nev
      Real Xn,V,r,Xo,Sum,X50,Error,p,p_last,Ywor,G,d_1,
     1     Xap,ht,theta,diam,A_F,B_F,C_F,D_F,E_F,F_F,U,
     2     d_2,n,n_1,n_2,CCS,T,Half_R,Q,Xim,dt
      Double Precision F,DF,L_1,L_2,C,B,Yim
      Go To (100,110,120,130,140,150,160,170,180),Nev
C     ******************************************************************
C     Set values due to crushing-For McCulley
C     ******************************************************************
 100   r=A(lent,3)
       n=0.668
       n_1=2.0449
       n_2=0.513
       G=0.20892
       dt=A(lent,2)
       CCS=65.0
       Xo=CCS*2.347
       d_2=Xo
       d_1=CCS*0.236
       Xim=Xo*0.5
       C=1-((Xim-d_2)/(d_1-d_2))**n
       B=G*((Xim/dt)**n_1)+(1-G)*((Xim/dt)**n_2)
       L_1=log(1.0-C*B-(1.0-C)*(1.0-(1.0-Xim/dt)**r))
       L_2=log(1-Xim/dt)
       p=L_1/L_2
      Call Seta(lent,3,p)
      Call Seta(lent,2,Xo)
c     ******************************************************************
c     Set values for the screen to classify-for double deck
c     ******************************************************************
      Xap=69.0
      theta=0.2269
      r=p
      diam=19.0
      U=120.0
      ht=(Xap+diam)*cos(theta)-diam
       If (ht.GE.50.8) then
         A_F = 0.3388*ht+14.4122
       Else
         A_F = 12.121286*(ht**0.3162)-10.2991
       Endif
      Q=((1.0-Xap/Xo)**r)
       If (Q.GT.0.87) then
         B_F=4.25*Q+4.275
       Else
         B_F=(-1.2)*Q+1.6
       Endif
      Half_R=(1.0-(1.0-Xap/(2.0*Xo))**r)*100.0
       If (Half_R.GE.80.0) C_F=0.05*Half_R-1.5
       If (Half_R.GE.55.0.and.Half_R.LT.80.0) C_F=0.0061*(Half_R**1.37)
       If (Half_R.GT.30.0.and.Half_R.LT.55.0) C_F=0.1528*(Half_R**0.564)
       If (Half_R.LE.30.0) C_F=0.012*Half_R+0.7
      D_F=1.0
      T=1.26*ht
       If (T.GT.32.0) then
        E_F=1.15
       else
        E_F=1.35-0.00625*T
       Endif
      F_F=U/1602.0
      X50=ht*((350*(1-Q))/(A_F*B_F*C_F*D_F*E_F*F_F*8.33))**(-0.148)
      CALL Seta(Lent,1,X50)
      Sum=0.0
      Xn=Xo
   10  If (Xn.GT.1) then
         V = (1.0-(Xn-1.0)/Xo)**r-(1.0-Xn/Xo)**r
         Sum=Sum+V*(Xn-0.5)
         Xn=Xn-1.0
         Go To 10
       Endif
      CALL SetA(lent,4,Sum)
      Go To 1000
c     ******************************************************************
c     Topsize from Double Deck Screen
c     ******************************************************************
  110 Xo=A(lent,2)
      X50=A(lent,1)
      Xim=0.65*Xo
      r=A(lent,3)
       C=1-exp(-0.693*(Xim/X50)**5.846)
       Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
       Yim=C*Ywor
       Error=2.5
       p=1.2
  300   If (ABS(Error).GT.0.0001) then
          p_last=p
          F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
          L_1=log(1-(Xim-0.5)/Xo)
          L_2=log(1-(Xim+0.5)/Xo)
          DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
          p=p-(F/DF)
          Error=ABS((p-p_last)/p)*1000
          Go To 300
        EndIf
       Call Seta(lent,3,p)
       Go To 1000
c     ****************************************************************:
c     Feed to bottom deck
c     ****************************************************************:
  120 Xo=A(lent,2)
      X50=A(lent,1)
      Xim=Xo*0.16666667
      r=A(lent,3)
       C=1-exp(-0.693*(Xim/X50)**5.846)
       Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
       Xo=69.0
       Yim=Ywor*(1-C)
       p=1.20
       Error=2.5
  310   If (ABS(Error).GT.0.0001) then
          p_last=p
          F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
          L_1=log(1-(Xim-0.5)/Xo)
          L_2=log(1-(Xim+0.5)/Xo)
          DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
          p=p-(F/DF)
          Error=ABS((p-p_last)/p)*1000
         Go To 310
       EndIf
      Call Seta(lent,3,p)
      Call Seta(lent,2,Xo)
      Xap=30.0
      theta=0.2269
      r=p
      diam=10.0
      U=320.0
      ht=(Xap+diam)*cos(theta)-diam
       If (ht.GE.50.8) then
         A_F = 0.3388*ht+14.4122
       Else
         A_F = 12.121286*(ht**0.3162)-10.2991
       Endif
      Q=((1.0-Xap/Xo)**r)
       If (Q.GT.0.87) then
         B_F=4.25*Q+4.275
       Else
         B_F=(-1.2)*Q+1.6
       Endif
      Half_R=(1.0-(1.0-Xap/(2.0*Xo))**r)*100.0
       If (Half_R.GE.80.0) C_F=0.05*Half_R-1.5
       If (Half_R.GE.55.0.and.Half_R.LT.80.0) C_F=0.0061*(Half_R**1.37)
       If (Half_R.GT.30.0.and.Half_R.LT.55.0) C_F=0.1528*(Half_R**0.564)
       If (Half_R.LE.30.0) C_F=0.012*Half_R+0.7
      D_F=1.0
      T=1.26*ht
       If (T.GT.32.0) then
        E_F=1.15
       else
        E_F=1.35-0.00625*T
       Endif
      F_F=U/1602.0
      X50=ht*((350*(1-Q))/(A_F*B_F*C_F*D_F*E_F*F_F*8.33))**(-0.148)
      CALL Seta(Lent,1,X50)
      Sum=0.0
      Xn=Xo
   20  If (Xn.GT.1) then
         V = (1.0-(Xn-1.0)/Xo)**r-(1.0-Xn/Xo)**r
         Sum=Sum+V*(Xn-0.5)
         Xn=Xn-1.0
         Go To 20
       Endif
      CALL SetA(lent,4,Sum)
      Go To 1000
c     ******************************************************************
c     Middelsize from Double Deck Screen
c     ******************************************************************
  130 Xo=A(lent,2)
      X50=A(lent,1)
      Xim=0.65*Xo
      r=A(lent,3)
       C=1-exp(-0.693*(Xim/X50)**5.846)
       Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
       Yim=C*Ywor
       Error=2.5
       p=1.2
  320   If (ABS(Error).GT.0.0001) then
          p_last=p
          F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
          L_1=log(1-(Xim-0.5)/Xo)
          L_2=log(1-(Xim+0.5)/Xo)
          DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
          p=p-(F/DF)
          Error=ABS((p-p_last)/p)*1000
          Go To 320
        EndIf
       Call Seta(lent,3,p)
       Go To 1000
c      *******************************************************************
c      Undersize from Double Deck Screen
c      *******************************************************************
  140  Xo=A(lent,2)
       X50=A(lent,1)
       Xim=Xo*0.16666667
       r=A(lent,3)
        C=1-exp(-0.693*(Xim/X50)**5.846)
        Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
        Xo=30.0
       Yim=Ywor*(1-C)
       p=1.20
       Error=2.5
  330  If (ABS(Error).GT.0.0001) then
          p_last=p
          F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
          L_1=log(1-(Xim-0.5)/Xo)
          L_2=log(1-(Xim+0.5)/Xo)
          DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
          p=p-(F/DF)
          Error=ABS((p-p_last)/p)*1000
         Go To 330
       EndIf
       Call Seta(lent,3,p)
       Call Seta(lent,2,Xo)
      Go To 1000
C     ******************************************************************
C     Set values due to crushing-For Simon 1&2
C     ******************************************************************
 150   r=A(lent,3)
       n=1.7846
       n_1=2.6816
       n_2=0.5413
       G=0.209
       dt=A(lent,2)
       CCS=25.0
       Xo=CCS*1.8553
       d_2=Xo
       d_1=CCS*1.04
       Xim=Xo*0.5
       C=1.0
       B=G*((Xim/dt)**n_1)+(1-G)*((Xim/dt)**n_2)
       L_1=log(1.0-C*B-(1.0-C)*(1.0-(1.0-Xim/dt)**r))
       L_2=log(1-Xim/dt)
       p=L_1/L_2
      Call Seta(lent,3,p)
      Call Seta(lent,2,Xo)
C     ******************************************************************
C     Set values for the screen to classify-Screens of simon 1&2
C     ******************************************************************
      Xap=37.5
      theta=0.2269
      r=p
      diam=8.9
      U=320.0
      ht=(Xap+diam)*cos(theta)-diam
       If (ht.GE.50.8) then
         A_F = 0.3388*ht+14.4122
       Else
         A_F = 12.121286*(ht**0.3162)-10.2991
       Endif
      Q=((1.0-Xap/Xo)**r)
       If (Q.GT.0.87) then
         B_F=4.25*Q+4.275
       Else
         B_F=(-1.2)*Q+1.6
       Endif
      Half_R=(1.0-(1.0-Xap/(2.0*Xo))**r)*100.0
       If (Half_R.GE.80.0) C_F=0.05*Half_R-1.5
       If (Half_R.GE.55.0.and.Half_R.LT.80.0) C_F=0.0061*(Half_R**1.37)
```

```
        If (Half_R.GT.30.0.and.Half_R.LT.55.0) C_F=0.1528*(Half_R**0.564)
        If (Half_R.LE.30.0) C_F=0.012*Half_R+0.7
       D_F=1.0
       T=1.26*ht
        If (T.GT.32.0) then
         E_F=1.15
        else
         E_F=1.35-0.00625*T
        Endif
       F_F=U/1602.0
       X50=ht*((350*(1-Q))/(A_F*B_F*C_F*D_F*E_F*F_F*8.33))**(-0.148)
       CALL Seta(Lent,1,X50)
       Sum=0.0
       Xn=Xo
   30   If (Xn.GT.1) then
          V = (1.0-(Xn-1.0)/Xo)**r-(1.0-Xn/Xo)**r
          Sum=Sum+V*(Xn-0.5)
          Xn=Xn-1.0
          Go To 30
        Endif
       CALL SetA(lent,4,Sum)
       Go To 1000
c      ****************************************************************:
c      Topsize from screen of Simon 1&2 Crusher
c      ****************************************************************:
  160  Xo=A(lent,2)
       X50=A(lent,1)
       Xim=0.5*Xo
       r=A(lent,3)
        C=1-exp(-0.693*(Xim/X50)**5.846)
        Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
        Yim=C*Ywor
        Error=2.5
        p=1.2
  340    If (ABS(Error).GT.0.0001) then
           p_last=p
           F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
           L_1=log(1-(Xim-0.5)/Xo)
           L_2=log(1-(Xim+0.5)/Xo)
           DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
           p=p-(F/DF)
           Error=ABS((p-p_last)/p)*1000
           Go To 340
         EndIf
        Call Seta(lent,3,p)
        Go To 1000
c      ****************************************************************:
c      Bottom size from screen of Simon 1&2 Crusher
c      ****************************************************************:
  170  Xo=A(lent,2)
       X50=A(lent,1)
       Xim=Xo*0.16666667
       r=A(lent,3)
        C=1-exp(-0.693*(Xim/X50)**5.846)
        Ywor=(1-(Xim-0.5)/Xo)**r-(1-(Xim+0.5)/Xo)**r
        Xo=37.5
        Yim=Ywor*(1-C)
        p=1.20
        Error=2.5
  350   If (ABS(Error).GT.0.0001) then
```

```
          p_last=p
          F=(1-(Xim-0.5)/Xo)**p-(1-(Xim+0.5)/Xo)**p-Yim
          L_1=log(1-(Xim-0.5)/Xo)
          L_2=log(1-(Xim+0.5)/Xo)
          DF=((1-(Xim-0.5)/Xo)**p)*L_1-((1-(Xim+0.5)/Xo)**p)*L_2
          p=p-(F/DF)
          Error=ABS((p-p_last)/p)*1000
         Go To 350
       EndIf
       Call Seta(lent,3,p)
       Call Seta(lent,2,Xo)
      Go To 1000
C     *******************************************************************
C     Set values due to crushing of Crushers 3&4
C     *******************************************************************
 180   r=A(lent,3)
       n=1.7846
       n_1=2.6816
       n_2=0.5413
       G=0.209
       dt=A(lent,2)
       CCS=25.0
       Xo=CCS*1.8553
       d_2=Xo
       d_1=CCS*1.04
       Xim=Xo*0.5
       C=1.0
       B=G*((Xim/dt)**n_1)+(1-G)*((Xim/dt)**n_2)
       L_1=log(1.0-C*B-(1.0-C)*(1.0-(1.0-Xim/dt)**r))
       L_2=log(1-Xim/dt)
       p=L_1/L_2
      Call Seta(lent,3,p)
      Call Seta(lent,2,Xo)
      Go To 1000
 1000 return
      end
c
c
```