

```

Interaction: '=====
              Traffic Flow Model INTAC-12 (SIMTOOL module)
              04.02.96 -----
for I=M to N
  J = I-1
  DX=X(J)-X(I)-L(J)
  DV=V(J)-V(I)
  if V(J)<=0 then
    SDXc=CC(0)
  else
    if DV>=0 or B(J)<=-1 then VIJ=V(I) else VIJ=V(J)-DV*(.5-Behav(I))
    SDXc=CC(0)+CC(1)*VIJ
  end if
  SDXo=CC(2)+SDXc
  SDV =CC(6)*DX^2
  if V(J)>0 then SDVc=CC(4)+SDV else SDVc=0
  if V(I)>CC(5) then SDVo=CC(5)+SDV else SDVo=SDV
  if DX<=SDXc and DV<=SDVo then
    OP$="A": Art(I)=3
    if V(I)>0 then
      if DV<0 then
        if DX>CC(0) then
          B(I)=min(B(J)+DV^2/(CC(0)-DX), B(I))
        else
          B(I)=min(B(J)+.5*(DV-SDVo), B(I))
        end if
        if B(I)>-CC(7) then
          B(I)=-CC(7)
        else
          B(I)=max(B(I), -10+.5*sqr(V(I)))
        end if
      end if
    else
      B(I)=0
    end if
  else if DV<SDVc and DX<SDXo+CC(3)*(DV-CC(4)) then
    OP$="B": Art(I)=2
    B(I)=.5*DV^2/(SDXc-DX-.01)
    B(I)=max(B(I), -10+sqr(V(I)))
  else if DV<SDVo and DX<SDXo then
    OP$="f": Art(I)=1
    if B(I)<=0 then
      B(I)=min(B(I), -CC(7))
    else
      B(I)=max(B(I), CC(7))
      if L(I)>=6.5 then B(I)=.5*B(I)
      B(I)=min(B(I), W(I)-V(I))
    end if
  else
    OP$="w": Art(I)=0
    if DX>SDXc then
      if right$(OP12$(I), 1) <> "w" then
        B(I)=CC(7)
      else
        Bmax=CC(8)+CC(9)*min(V(I), 22.2)+Behav(I)
        if DX<SDXo then
          B(I)=min(DV^2/(SDXo-DX), Bmax)
        else
          B(I)=Bmax
        end if
      end if
      if L(I)>=6.5 then B(I)=.5*B(I)
      B(I)=min(B(I), W(I)-V(I))
    else
      B(I)=0
    end if
  end if
  OP12$(I)=right$(OP12$(I), 11)+OP$
next
return '-----

```

Verwendung CC(6) und CC(9)
siehe Seite 2

75.000

Bruss AX

} 4

was bei $V > 0$ zu vernünftl. an

$$SDV = \frac{DX - SDXo}{CC3} + CC4$$

SDXV

Bruss AX

↳ "Bruss" (f)

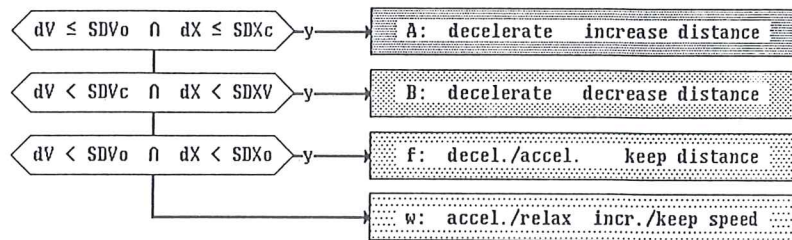
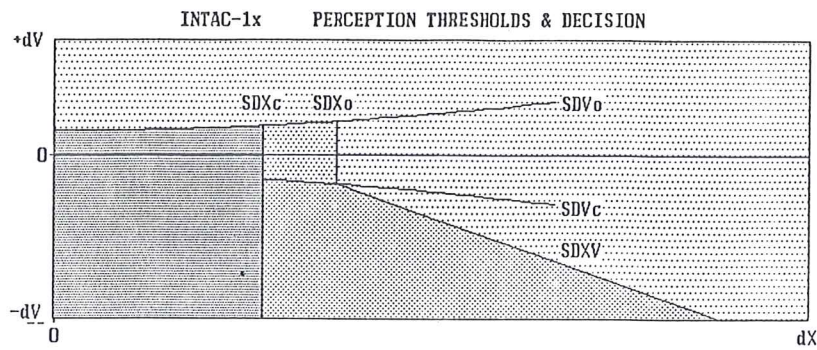
2.5.98

↳ wie zu 2.5.98?

Bruss

und dann kein Split (wg immer noch "Frei"), hmmm...

Modellierung des Lkw ist sehr grob!



SIMTOOL	INTAC-12-Constants	Calibration: 15.02.98 - 13:50
THRESHOLDS for dX	SDXc [minimal distance at V=0 [m] : 1.50 [additional time headway at V>0 [s] : 1.30 SDXo maximal drift from SDXc at dV=0 [m] : 4.00 SDXV SDXo + (-dX/dV) at dV<0 [s] : 12.00	
THRESHOLDS for dV	SDVc [minimal closing dV at dV<0 [m/s] : -0.25 SDVo [minimal opening dV at dV>0 [m/s] : 0.35 ± dV/dX [1/10^4 rad/s] : 6.00	
DRIVING ACTIVITIES	car following activities ± b [m/s²] : 0.25 acceleration behaviour when starting [m/s²] : 2.00 acceleration behaviour at V=80 km/h [m/s²] : 1.50	

CC

0
1
2
3
4
5
6
7
8
9

```

SystemStatus: '.....
for I=M to N
  XV(I)=X(I)
  V(I)=V(I)
  per=X(I-1)-L(I-1)
  - B(I)<-V(I) then
    X(I)=XV(I)-.5*VV(I)^2/B(I)
    V(I)=0
    if Art(I)<2 then Art(I)=3: mid$(OP12$(I),12,1)=chr$(254)
  else
    X(I)=XV(I)+VV(I)+.5*B(I)
    V(I)=VV(I)+B(I)
  end if
  if X(I)>Bumper+.01 then
    X(I)=Bumper: V(I)=V(I-1)
    if X(I)>150 then mid$(OP12$(I),12,1)=chr$(219)
  end if
next
return '.....
  
```

$$CC(6) = 0.0001 * CC(6) / 1.2$$

$$CC(9) = (CC(9) - CC(8)) * 3.6 / 80$$

Grid nach Anfang
von CALIBRATION bzw. 1s.
so umgewandelt, um
Rechenzeit zu sparen!

```
sub Calibration(CC(1),IM$,CalibTime$)
' SIMTOOL-Module 14.01.97
```

Relevante Teile

```
dim CC$(15)
CC$(0) = "THRESHOLDS"
CC$(1) = " for dX"
CC$(2) = "
CC$(3) = "
CC$(4) = "THRESHOLDS"
CC$(5) = " for dV"
CC$(6) = "
CC$(7) = "DRIVING"
CC$(8) = "ACTIVITIES"
CC$(9) = "
NC=9
```

SDXc	[minimal distance	at V=0	[m]	"
	[additional time headway	at V>0	[s]	"
SDXo	maximal drift from SDXc	at dV=0	[m]	"
SDXV	SDXo + (-dX/dV)	at dV<0	[s]	"
SDVc	[minimal closing dV	at dV<0	[m/s]	"
SDVo	[minimal opening dV	at dV>0	[m/s]	"
	± dV/dX	[1/10 ⁻⁴ rad/s]	"	"
	car following activities	± b	[m/s ²]	"
	acceleration behaviour when starting		[m/s ²]	"
	acceleration behaviour at V=80 km/h		[m/s ²]	"

CalibDataChange:

```
color 12: locate 25,7: print "<d>: Calibration Data"
while not instat: wend
color 15: locate 25,1: print space$(79);
AC$=inkey$
if AC$="d" then
  RRR=5
  call Indat(RRR+ 0,68,"###.##",CC(0),0,20)
  call Indat(RRR+ 1,68,"###.##",CC(1),.5,2.5)
  call Indat(RRR+ 2,68,"###.##",CC(2),0,10)
  call Indat(RRR+ 3,68,"###.##",CC(3),-99,99)
  call Indat(RRR+ 5,68,"###.##",CC(4),-2,2)
  call Indat(RRR+ 6,68,"###.##",CC(5),0,3)
  call Indat(RRR+ 7,68,"###.##",CC(6),0,20)
  call Indat(RRR+ 9,68,"###.##",CC(7),-1,1)
  call Indat(RRR+10,68,"###.##",CC(8),1,8)
  call Indat(RRR+11,68,"###.##",CC(9),.5,CC(8))
  for I=0 to NC
    CC(I)=abs(CC(I))
  next
  CC(3)=-CC(3)
  CC(4)=-CC(4)
```

*Versendung
CC(6) und CC(9)
siehe unten*

end sub

```
SimRunInit:
MaxN=350
erase W,L,Behav,X,XV,V,VV,B,Art,OP12$,G,GG
dim W(350),L(350),Behav(350)
dim X(350),XV(350),V(350),VV(350),B(350),Art(350),OP12$(350)
dim G(11000),GG(11000)
CC(6)=.0001*CC(6)/1.7
CC(9)=(CC(9)-CC(8))*3.6/80
return
```



```

Generator: '-----
while T>=TI
  if X(N)<=-50 then
    TI=T+1.00001
  else
    N=N+1
    if N>MaxN then
      Spe=M-1
      for I=M to N
        J=I-Spe
        W(J)=W(I): L(J)=L(I): Behav(J)=Behav(I)
        X(J)=X(I): XV(J)=XV(I): V(J)=V(I): VV(J)=VV(I): B(J)=B(I)
        Art(J)=Art(I): OP12$(J)=OP12$(I)
      next
      M=M-Spe: N=N-Spe: Qinc=Qinc-Spe: S1=S1-Spe: S2=S2-Spe: S12=S12-Spe
      for I=N+1 to MaxN
        X(I)=5000: XV(I)=5000
      next
    end if

    if EpsW>0 then
      Behav(N)=.5*RandTin
      if RandTin<.15 then
        W(N)=W00*SW*(RandTin)/.15
      elseif RandTin<.85 then
        W(N)=W15+2*SW*(RandTin-.15)/.7
      else
        W(N)=W85+SW*(RandTin-.85)/.15
      end if
    else
      Behav(N)=0
      W(N)=WQ
    end if

    RandTrk=10*RandTin: RandTrk=RandTrk-int(RandTrk)
    if RandTrk<.01*Trk then
      L(N)=10
      if EpsW>0 then W(N)=.9*W(N): if W(N)>25 then W(N)=25
    else
      if EpsW>0 then L(N)=3.70+1.295*RandTrk else L(N)=4.35
    end if

    XVzul=XV(N-1)-L(N-1)-CC(0)-CC(1)*W(N)
    XV(N)=W(N)*(T-1-TI)
    if XV(N)<=XVzul then
      V(N)=W(N)
      Art(N)=0
    else
      XV(N)=XVzul
      if V(N-1)<W(N) then V(N)=V(N-1) else V(N)=W(N)
      Art(N)=3
    end if
    X(N)=XV(N)+V(N)
    OP12$(N)=space$(11)+" "

    if EpsW=0 then
      Z0=ZZ
    elseif V(N)>1 then
      Z0=(L(N)+CC(0)+.5*CC(2))/V(N)+CC(1)
    end if
    RandTin=rnd
    if ZZ>Z0 then TI=TI+Z0-(ZZ-Z0)*log(1-RandTin) else TI=TI+Z0
  end if
wend
return '-----

```

```

Deletor: '-----
while X(M)>XE+50
  X(M)=X(M)+51
  M=M+1
  if X(M)>XE+25.01 then W(M)=.6*V(M)+.4*W(M)
wend
return '-----

```

```

Incidents: '-----
if T mod TincA=0 then
  if Incid=1 then
    TincA=99999
    TincE=T+TincD
    BRinc=-.125          '=-.5/4.0 (incident)
  elseif Incid=2 then
    TincE=T+TincD
    BRinc=-.25          '=-.5/2.0 (traffic light)
  end if
  Qinc=M
  while X(Qinc)>=Xinc-BRinc*(Winc^2-V(Qinc)^2)
    Qinc=Qinc+1
  wend
end if

if X(Qinc)<Xinc and X(Qinc)>Xinc-200 and V(Qinc)>=Winc then
  Binc=.5*(Winc^2-V(Qinc)^2)/(Xinc-X(Qinc)-1)
else
  Binc=0
end if

if T<=TincE then
  if Winc=0 then
    B(Qinc)=min(B(Qinc),Binc)
  else
    B(Qinc)=min(B(Qinc),max(Binc,Winc-V(Qinc)))
  end if
  if B(Qinc)<0 then Art(Qinc)=3: mid$(OP12$(Qinc),12,1)=chr$(170)
end if
return '-----

```

```

SystemStatus: '-----
for I=M to N
  XV(I)=X(I)
  VV(I)=V(I)
  Bumper=X(I-1)-L(I-1)
  if B(I)<=V(I) then
    X(I)=XV(I)-.5*VV(I)^2/B(I)
    V(I)=0
    if Art(I)<2 then Art(I)=3: mid$(OP12$(I),12,1)=chr$(254)
  else
    X(I)=XV(I)+VV(I)+.5*B(I)
    V(I)=VV(I)+B(I)
  end if

  if X(I)>Bumper+.01 then
    X(I)=Bumper: V(I)=V(I-1)
    if X(I)>150 then mid$(OP12$(I),12,1)=chr$(219)
  end if
next
return '-----

```

```

MacroInit: '-----
M=1: N=0
X(0)=2200: XV(0)=X(0): V(0)=WQ 'Phantom: XE+200 m
randomize R
RandTin=rnd
TI=.00001
return '-----

```