DDE workaround for time-varying initial condition

Simple example

- Following example $\begin{cases} \frac{dG}{dt} = -k * delay(G, \tau) \\ G(t) = a * \exp(b * t) \; for \; t < 0 \end{cases}$
- Initial implementation in [LONGITUDINAL]

EQUATION:
; initialization of the time
t0 = 0
; initialization of the variable of interest
G_0 = a*exp(b*t)
; dynamics
ddt_G = -k*delay(G,tau)

Simple example workaround

- Workaround implementation

```
EQUATION:
t0 = -tau
; initialization of the variable of interest
G_0 = a*exp(b*t0)
dG_ic = a*exp(b*t)*b

ic_on = 0
if t<0
  ic_on = 1
end

ddt_G = (1-ic_on)*(-k*delay(G,tau)) + ic_on*dG_ic
```

the initial time must be initialized to a value less than -tau

The variable is initialized at t0
We compute the derivative of the IC

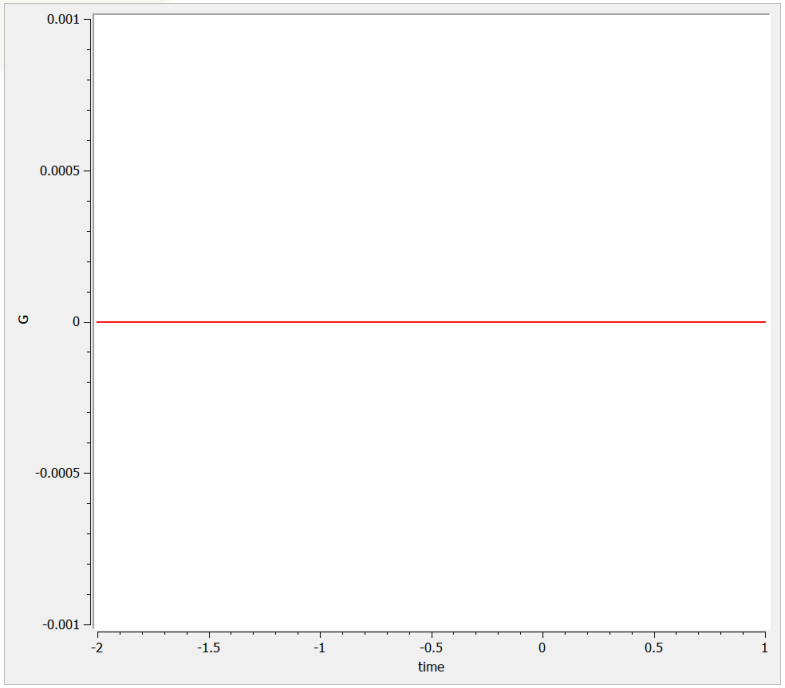
ic_on is a "boolean" saying if the initial condition should be taken into account

Modification of the dynamics to switch between conditions

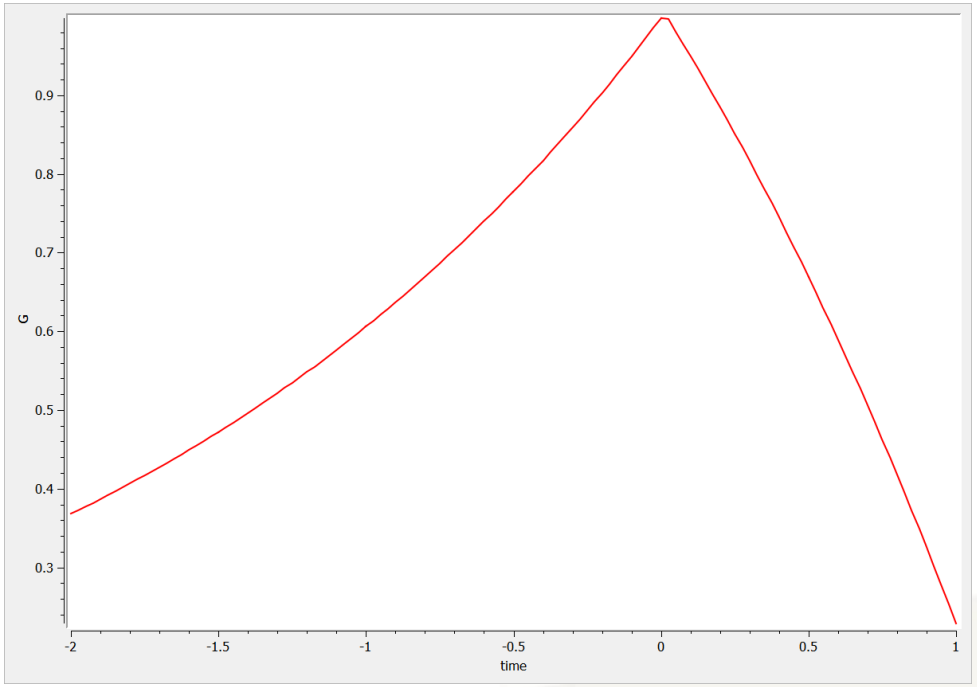
Simple example workaround

- In Mlxplore 2016R1 (modelDelay without and with workaround)

modelDelay.mlxplore.mlxtran



modelDelayWorkaround.mlxplore.mlxtran



Arthritis case

- The same workaround works for more complex models
- The several steps are
  - Modify the initial time to a value less than -tau
  - Add the derivative of the initial condition for the time varying conditions
  - Define the ic_on
  - Modify the dynamics to take the ic_on into account

arthritis_modelWO.txt

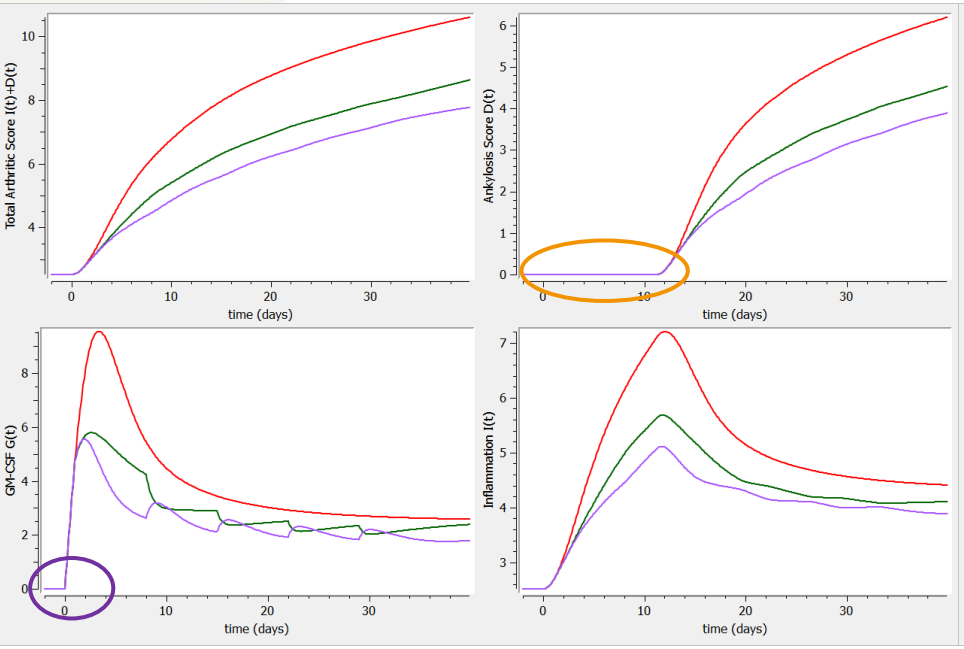
```
;DESCRIPTION: arthritis development model of Koch

[LONGITUDINAL]
input = {a, b, I0, alpha, beta, CL, V1, V2, sigma1, sigma2, sigma3, k1, k2, k3, k4, k5, tau}

EQUATION:
; initialization of the time
t0 = -2*tau

; initialization of the variable of interest
I_0 = I0
D_0 = 0
G_0 = a*exp(b*t0)
dG = a*exp(b*t)*b
ic_on = 0
if t<0
  ic_on = 1
end

K12 = alpha*beta*V2/CL
K21 = alpha*beta*V1/CL
Cc = pkmodel(k12=K12,k21=K21,V=V1,Cl=CL)
E = (sigma1*exp(- sigma2*Cc) + sigma3)*Cc

ddt_G = (1-ic_on)*(k3 - E*G - (k1/k2)*(1- exp(- k2*t))*G)+ic_on*dG_ic
ddt_I = (1-ic_on)*(k4*G - k4*delay(G,tau))
ddt_D = (1-ic_on)*(- k5*D + k4*delay(G,tau))

TAS = I+D
AKS = D

[INDIVIDUAL]
input = {tau_pop, omega_tau}

DEFINITION:
tau = {distribution = normal, mean = tau_pop, sd = omega_tau}
```

Notice that as G is the only one with a time varying condition, it is not necessary to add something to the other dynamics

arthritis_modelWO.txt

- In Mlxplore 2016R1 (arthrithis.mlxplore.mlxtran without and with workaround)

arthrithis_model.txt



arthrithis_modelWorkaround.txt