```
CONTEXT
   System_Ctx
CONSTANTS
   S
   TIME
   sigma
AXIOMS
   axm1   :   S=RReal×RReal
   axm2   :   TIME=RRealPlus
   axm3   :   sigma∈ RRealPlus ∧ sigma ↦Rzero ∈gt
END
```

```
CONTEXT
   Theorems
AXIOMS
   axm1   :   ∀a,b,c,d·a↦b∈ leq ∧ c↦d ∈leq ⇒ plus(a↦c) ↦plus(b↦d) ∈ leq
   axm2   :   ∀a,b,c,d·Rzero↦a ∈ leq ∧ Rzero↦b ∈ leq ∧ Rzero↦c ∈ leq ∧ Rzero↦d ∈ leq ∧ a↦b∈ leq ∧ c↦d ∈leq ⇒ times
              (a↦c) ↦times(b↦d) ∈ leq
   axm3   :   ∀a,b,c· a↦b ∈ leq ∧ b↦c ∈ leq⇒ a ↦c ∈ leq
              ∀a,b· a∈ RReal ∧ b ∈ RReal
   axm4   :   ⇒
              minus(times(a↦a) ↦times(b↦b))=times(plus(a↦b)↦minus(a↦b))
   axm5   :    ∀a· a∈ RReal ⇒ uminus(a)=minus(Rzero↦a)
              ∀a· a∈ RReal ⇒
              a=plus(
                     times(divide(Rone ↦ Rtwo) ↦a)
   axm6   :      ↦
                     times(divide(Rone ↦ Rtwo) ↦a)
                   )
              ∀a,b· a∈ RReal ∧ b∈ RReal ∧ times(a↦b)∈ RRealStar
   axm7   :   ⇒
              inverse(times(a↦b))=times(inverse(a)↦inverse(b))
END
```

```
MACHINE
    System_M
SEES
    System_Ctx
    Theorems
VARIABLES
    t
    plantV
INVARIANTS
    inv1   :   t ∈ TIME
    inv2   :   plantV ∈ Closed2Closed(Rzero, t) ⇸ S
EVENTS
    INITIALISATION   ≙
    STATUS
      ordinary
    BEGIN
      act1   :   t≔Rzero
      act2   :   plantV :∈ {Rzero} ⟶ S
    END

    Progress   ≙
    STATUS
      ordinary
    BEGIN
      act1   :     t :| t' ∈ TIME ∧ (t ↦ t' ∈ lt ∧ minus(t'↦t) ↦ sigma ∈ geq)
    END

    Plant   ≙
    STATUS
      ordinary
    ANY
      e
      plant1
    WHERE
      grd1   :   e ∈ DE(S)
      grd2   :   Solvable(Closed2Closed(Rzero, t)\dom(plantV),e)
                 plant1 ∈ Closed2Closed(Rzero, t)\dom(plantV) ⟶ S ∧
      grd3   :   AppendSolutionBAP(e,
                 Closed2Closed(Rzero, t)\dom(plantV),
                 Closed2Closed(Rzero, t)\dom(plantV), plant1)
    THEN
      act1   :   plantV≔plantV⩤plant1
    END

END
```

```
CONTEXT
    EventTriggered_Ctx
EXTENDS
    System_Ctx
SETS
    EXEC
    PROP
CONSTANTS
    safe
    evt_trig
    ctrl
    plant
    prg
    f_evol
    f_evol_plantV
    evade_value
AXIOMS
    axm1   :   safe ∈ (S × RReal) ⟶ BOOL
    axm2   :   evt_trig ∈ S × RReal ×RReal ⟶ BOOL
    axm3   :   partition(EXEC,  {ctrl},{plant},{prg})
    axm4   :   f_evol ∈ RReal ⟶ S
    axm5   :   f_evol_plantV ∈ (RReal ⟶ (TIME × S ⟶ (RReal×RReal)))
    axm6   :   ∀ ctrlV · ctrlV ∈ RReal ⇒ (f_evol_plantV(ctrlV) =
                      (λ t ↦ plantV · t ∈ TIME ∧ plantV ∈ S | f_evol(ctrlV)))
    axm7   :   evade_value⊆RReal ∧ evade_value≠∅
END
```

```
MACHINE
   EventTriggered_M
REFINES
   System_M
SEES
   EventTriggered_Ctx
VARIABLES
   t
   plantV
   ctrlV
   exec
INVARIANTS
   inv1   :   ctrlV ∈ RReal
   inv2   :   exec ∈ EXEC
   inv3   :   exec≠plant ⇒ dom(plantV)=Closed2Closed(Rzero, t)
   inv4   :   exec=plant ⇒ t∉ dom(plantV)
EVENTS
   INITIALISATION  ≙
     extended
   STATUS
     ordinary
   BEGIN
     act1   :   t≔Rzero
     act2   :   plantV :∈ {Rzero} ⟶ S
     act3   :   ctrlV :∈ RReal
     act4   :   exec ≔ ctrl
   END

   Progress  ≙
   STATUS
     ordinary
   REFINES
     Progress
   ANY
     t1
   WHERE
     grd1   :   exec=prg
     grd2   :   t1 ∈ TIME ∧ (t ↦ t1 ∈ lt ∧ minus(t1↦t) ↦ sigma ∈ geq)
     grd3   :   ctrlV∉ evade_value⇒evt_trig(plantV(t)↦minus(t1↦t)↦ctrlV) = TRUE
   THEN
     act1   :   t≔t1
     act2   :   exec ≔ plant
   END

   Plant  ≙
   STATUS
     ordinary
   REFINES
     Plant
   ANY
     plant1
   WHERE
     grd1   :   exec=plant
     grd2   :   plant1 ∈ Closed2Closed(Rzero, t)\dom(plantV) ⟶ S
     grd3   :   ode(f_evol_plantV(ctrlV),plant1(t),t) ∈ DE(S)
     grd4   :   Solvable(Closed2Closed(Rzero, t)\dom(plantV),
                       ode(f_evol_plantV(ctrlV),plant1(t),t))
                AppendSolutionBAP(ode(f_evol_plantV(ctrlV),plant1(t),t),
     grd5   :   Closed2Closed(Rzero, t)\dom(plantV),
                Closed2Closed(Rzero, t)\dom(plantV), plant1)
   WITH
     e   :   e = ode(f_evol_plantV(ctrlV),plant1(t),t)
   THEN
     act1   :   plantV≔plantV⊲plant1
     act2   :   exec≔ctrl
   END

   Ctrl_normal  ≙
   STATUS
```

```
    ordinary
  ANY
    nrml_value
  WHERE
    grd1   :   exec = ctrl
    grd2   :   nrml_value∈RReal
    grd3   :   nrml_value∉ evade_value ⇒safe(plantV(t)↦nrml_value) = TRUE
  THEN
    act1   :   ctrlV ≔nrml_value
    act2   :   exec ≔ prg
  END

  Ctrl_evade  ≙
  STATUS
    ordinary
  ANY
    evade_val
  WHERE
    grd1   :   exec = ctrl
    grd2   :   evade_val∈evade_value
  THEN
    act1   :   ctrlV≔ evade_val
    act2   :   exec ≔ prg
  END

END
```

```
CONTEXT
    Car_Event_Ctx
EXTENDS
    EventTriggered_Ctx
CONSTANTS
    A
    B
    SP
    pinit
    vinit
AXIOMS
    axm1   :   A ∈ RReal ∧ Rzero ↦  A ∈ lt
    axm2   :   B ∈ RReal ∧ Rzero ↦  B ∈ lt ∧ evade_value={uminus(B),Rzero}
    axm3   :   SP ∈ RReal
    axm4   :   Rzero ↦  SP ∈ lt
    axm5   :   pinit∈ RRealPlus ∧ pinit↦SP∈leq
    axm6   :   vinit∈ RRealPlus
    axm7   :   safe = (λ (p↦v)↦ctrlV · (p↦v) ∈ S ∧ ctrlV ∈ RReal |
                         bool((plus(p↦ divide(times(v↦v) ↦times(Rtwo ↦ B))) ↦ SP ∈lt )))
               evt_trig = (λ (p↦v)↦t1↦ctrlV · (p↦v) ∈ S ∧ ctrlV ∈ RReal |
                         bool((
                                      plus(
               plus(
                    plus(
                         p ↦
                         times(divide(Rone ↦ Rtwo) ↦
    axm8   :                    times(ctrlV ↦ times(t1 ↦ t1)))
                         )
                    ↦
                    times(v ↦ t1)
                   )
               ↦
                                        divide(times(v↦v) ↦times(Rtwo ↦ B))) ↦ SP ∈leq) ) )
    axm9   :   plus(pinit↦ divide(times(vinit↦vinit) ↦times(Rtwo ↦ B))) ↦ SP ∈leq
    axm10   :   ∀ ctrlV · ctrlV ∈ RReal ⟹ (f_evol_plantV(ctrlV) =
                (λ t↦ (p↦v) · t ∈ TIME ∧ (p↦v) ∈ S |(v↦ctrlV)))
END
```

```
MACHINE
    Car_Event_M
REFINES
    EventTriggered_M
SEES
    Car_Event_Ctx
VARIABLES
    t
    ctrlV
    exec
    p
    v
INVARIANTS
    inv1   :   p ∈ Closed2Closed(Rzero, t) ⇸→ RReal
    inv2   :   v ∈ Closed2Closed(Rzero, t) ⇸→ RRealPlus
    inv3   :   exec≠plant ⇒ dom(p)=Closed2Closed(Rzero, t) ∧ dom(v)=Closed2Closed(Rzero, t)
    inv4   :   dom(v)=dom(p)
    inv5   :   plantV=bind(p,v)
    inv6   :   ∀x· x∈ dom(p)⇒ p(x)↦SP∈ leq
    inv7   :   exec=plant ⇒ t∉dom(plantV)
               ∀t1,t2· t1∈TIME ∧ t2∈TIME ∧
    inv8   :    dom(p)=Closed2Closed(Rzero,t1) ∧ dom(p)=Closed2Closed(Rzero,t2)
               ⇒
               t1=t2
EVENTS
    INITIALISATION   ≙
    STATUS
      ordinary
    WITH
      plantV'   :   plantV'=bind(p',v')
    BEGIN
      act1   :   t≔Rzero
      act2   :   p≔{Rzero↦pinit}
      act3   :   v≔{Rzero↦vinit}
      act4   :   ctrlV :∈ RReal
      act5   :   exec ≔ ctrl
    END

    Progress   ≙
    STATUS
      ordinary
    REFINES
      Progress
    ANY
      t1
    WHERE
      grd1   :   exec=prg
      grd2   :   t1 ∈ TIME ∧ (t ↦ t1 ∈ lt ∧ minus(t1↦t) ↦ sigma ∈ geq)
      grd3   :   ctrlV∉ evade_value⇒evt_trig((bind(p,v))(t)↦minus(t1↦t)↦ctrlV) = TRUE
    THEN
      act1   :   t≔t1
      act2   :   exec ≔ plant
    END

    Plant_event_car   ≙
    STATUS
      ordinary
    REFINES
      Plant
    ANY
      p1
      v1
    WHERE
      grd1   :   exec = plant
                 p1 ∈ Closed2Closed(Rzero, t)\dom(p) ⟶ RReal ∧
      grd2   :   v1 ∈ Closed2Closed(Rzero, t)\dom(v) ⟶ RRealPlus
      grd3   :   ode(f_evol_plantV(ctrlV),(p1(t)↦v1(t)),t) ∈ DE(S)
                 Solvable(Closed2Closed(Rzero, t)\dom(bind(p,v)),
      grd4   :                ode(f_evol_plantV(ctrlV),bind(p1,v1)(t),t))
```

```
  grd5   :   AppendSolutionBAP(ode(f_evol_plantV(ctrlV),(bind(p1,v1))(t),t),
             Closed2Closed(Rzero, t)\dom(bind(p,v)),
             Closed2Closed(Rzero, t)\dom(bind(p,v)), bind(p1,v1))
  grd6   :   ∀xx· xx∈ dom(p1)⇒ p1(xx)↦SP ∈ leq
WITH
  plant1   :   plant1=bind(p1,v1)
THEN
  act1   :   p≔p⩤p1
  act2   :   v≔v⩤v1
  act3   :   exec≔ctrl
END

Ctrl_Acceleration_car   ≙
STATUS
  ordinary
REFINES
  Ctrl_normal
WHEN
  grd1   :   exec = ctrl
  grd2   :   safe((bind(p,v))(t)↦A) = TRUE
WITH
  nrml_value   :   nrml_value=A
THEN
  act1   :   ctrlV ≔A
  act2   :   exec ≔ prg
END

Ctrl_Deceleration_car   ≙
STATUS
  ordinary
REFINES
  Ctrl_evade
ANY
  evade_val
WHERE
  grd1   :   exec = ctrl
  grd2   :   evade_val ∈ evade_value
  grd3   :   v(t)↦Rzero ∈ gt ⇒ evade_val=uminus(B)
  grd4   :   v(t)=Rzero ⇒evade_val=Rzero
THEN
  act1   :   ctrlV ≔ evade_val
  act2   :   exec ≔ prg
END

END
```

```
CONTEXT
    Car_Time_Ctx
EXTENDS
    Car_Event_Ctx
CONSTANTS
    epsilon
    safeEpsilon
AXIOMS
    axm1   :   epsilon ∈ TIME  ∧  sigma↦epsilon ∈leq
    axm2   :   safeEpsilon ∈ (S × RReal) ⟶ BOOL
               safeEpsilon = (λ (p↦v)↦ctrlV · (p↦v) ∈ S ∧ ctrlV ∈ RReal |
               bool(
               plus(
                    plus(p ↦ plus(times(v↦ epsilon)↦
                         times(divide(Rone ↦ Rtwo) ↦  times(A ↦ times(epsilon ↦ epsilon)))))
                   ↦
                   plus(
    axm3   :             plus (divide(times(v↦v)↦ times(Rtwo ↦ B))
                               ↦
                               divide(times(times(A ↦ A) ↦ times(epsilon ↦ epsilon)) ↦ times(Rtwo ↦ B)))
                         ↦
                         divide(times(A ↦ times(epsilon ↦ v)) ↦ B)
                         )
                )
               ↦ SP ∈ lt))
    axm4   :   Rzero↦epsilon ∈ lt
END
```

```
MACHINE
   Car_Time_M
REFINES
   Car_Event_M
SEES
   Car_Time_Ctx
   Theorems
VARIABLES
   t
   ctrlV
   exec
   p
   v
INVARIANTS
   inv1   :   ctrlV∈{Rzero,uminus(B),A}
              ∃ t1·t1 ∈TIME ∧ dom(p)=Closed2Closed(Rzero,t1) ∧
                  minus(t↦t1)↦epsilon ∈leq ∧
   inv2   :       (exec≠plant ⇒ t1=t) ∧
                  (exec=plant⇒ t↦t1∈gt) ∧
                  (ctrlV∉evade_value ∧ exec=plant  ⇒ safeEpsilon((p(t1)↦v(t1))↦A) = TRUE)
              ∀ t1· (t1 ∈TIME ∧ dom(p)=Closed2Closed(Rzero,t1)
              ⇒
              plus(
                     p(t1) ↦
                     divide(
                             times(v(t1)↦ v(t1))
   inv3   :                 ↦
                             times(Rtwo ↦ B)
                            )
                 )
               ↦ SP ∈ leq
              )
   inv4   :   ctrlV∉evade_value ∧ exec=prg  ⇒ safeEpsilon((p(t)↦v(t))↦A) = TRUE
   inv5   :   ∀ t1·t1 ∈TIME ∧ dom(p)=Closed2Closed(Rzero,t1) ∧
              ctrlV=Rzero ∧ exec≠ctrl ⇒ v(t1)=Rzero
EVENTS
   INITIALISATION   ≙
   STATUS
     ordinary
   BEGIN
     act1   :   t≔Rzero
     act2   :   p≔{Rzero↦pinit}
     act3   :   v≔{Rzero↦vinit}
     act4   :   ctrlV ≔ Rzero
     act5   :   exec ≔ ctrl
   END

   Progress_time   ≙
   STATUS
     ordinary
   REFINES
     Progress
   ANY
     t1
   WHERE
     grd1   :   exec=prg
     grd2   :   t1 ∈ TIME ∧ (t ↦ t1 ∈ lt ∧ minus(t1↦t) ↦ sigma ∈ geq)
     grd3   :   minus(t1↦t) ↦ epsilon ∈ leq
   THEN
     act1   :   t≔t1
     act2   :   exec ≔ plant
   END

   Plant_event_car   ≙
   STATUS
     ordinary
   REFINES
     Plant_event_car
   ANY
     p1
```

```
  v1
  lastTime
  epsilon1
WHERE
  grd1    :   exec = plant
              ∀t1,t2· t1∈TIME ∧ t2∈TIME ∧
               dom(p)=Closed2Closed(Rzero,t1) ∧ dom(p)=Closed2Closed(Rzero,t2)
  grd2    :   ⇒
              t1=t2
  grd3    :   lastTime∈ TIME ∧ dom(p)=Closed2Closed(Rzero, lastTime)
  grd4    :   lastTime∈dom(p)
  grd5    :    lastTime∈dom(v)
              ctrlV=uminus(B)  ⇒
              (minus(t↦lastTime)↦ divide(v(lastTime)↦B)∈leq ⇒epsilon1=minus(t↦lastTime))
  grd6    :   ∧
              (minus(t↦lastTime)↦ divide(v(lastTime)↦B)∈gt⇒epsilon1=divide(v(lastTime)↦B))
  grd7    :   ctrlV∈{Rzero,A}  ⇒ epsilon1=minus(t↦lastTime)
              p1= (λ t1 · t1 ∈ RReal ∧ t1↦ lastTime ∈ gt ∧  t1 ↦ t ∈ leq |
              plus(
                   plus(
                        p(lastTime) ↦
                        times(divide(Rone ↦ Rtwo) ↦
  grd8    :                    times(ctrlV ↦ times(epsilon1 ↦ epsilon1)))
                        )
                   ↦
                   times(v(lastTime) ↦ epsilon1)
                  )
              )
              v1=(λ t1 · t1 ∈ RReal ∧ t1↦ lastTime ∈ gt ∧  t1 ↦ t ∈ leq|
                    plus(
                         times(ctrlV ↦ epsilon1)
  grd9    :                ↦
                         v(lastTime)
                        ))
  grd10   :    ode(f_evol_plantV(ctrlV),(p1(t)↦v1(t)),t) ∈ DE(S)
               Solvable(Closed2Closed(Rzero, t)\dom(bind(p,v)),
  grd11   :                 ode(f_evol_plantV(ctrlV),bind(p1,v1)(t),t))
               solutionOf(
                Closed2Closed(Rzero, t)\dom(bind(p,v)),
  grd12   :    (Closed2Closed(Rzero, t)\dom(bind(p,v))) ◁ bind(p1,v1),
                            ode(f_evol_plantV(ctrlV), bind(p1,v1)(t), t)
               )
THEN
  act1    :   p≔p◁p1
  act2    :   v≔v◁v1
  act3    :   exec≔ctrl
END

Ctrl_Acceleration_car_time   ≙
STATUS
  ordinary
REFINES
  Ctrl_Acceleration_car
WHEN
  grd1    :   exec = ctrl
  grd2    :   safeEpsilon((p(t)↦v(t))↦A) = TRUE
THEN
  act1    :   ctrlV ≔A
  act2    :   exec ≔ prg
END

Ctrl_Deceleration_car   ≙
STATUS
  ordinary
REFINES
  Ctrl_Deceleration_car
ANY
  evade_val
WHERE
  grd1    :   exec = ctrl
  grd2    :   evade_val ∈ evade_value
  grd3    :   v(t)↦Rzero ∈ gt ⇒ evade_val=uminus(B)
```

```
      grd4   :   v(t)
                 =Rzero ⇒evade_val=Rzero
    THEN
      act1   :   ctrlV ≔ evade_val
      act2   :   exec ≔ prg
    END

END
```