```
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
   axm4   :  ∀a,b· a∈ RReal ∧ b ∈ RReal
             ⇒
             minus(times(a↦a) ↦times(b↦b))=times(plus(a↦b)↦minus(a↦b))
   axm5   :  ∀a· a∈ RReal ⇒ uminus(a)=minus(Rzero↦a)
   axm6   :  ∀a· a∈ RReal ⇒
             a=plus(
                    times(divide(Rone ↦ Rtwo) ↦a)
                ↦
                    times(divide(Rone ↦ Rtwo) ↦a)
                  )
   axm7   :  ∀a,b· a∈ RReal ∧ b∈ RReal ∧ times(a↦b)∈ RRealStar
             ⇒
             inverse(times(a↦b))=times(inverse(a)↦inverse(b))
END
```

```
CONTEXT
   System_Ctx
CONSTANTS
   S
   TIME
   plantVInit
   sigma
AXIOMS
   axm1   :   S=RReal
   axm2   :   TIME=RRealPlus
   axm3   :   plantVInit∈S
   axm4   :   sigma∈ RRealPlus ∧ sigma ↦Rzero ∈gt
END
```

```
MACHINE
   System_M
SEES
   System_Ctx
VARIABLES
   t
   plantV
INVARIANTS
   inv1   :   t ∈ TIME
   inv2   :   plantV ∈ Closed2Closed(Rzero, t) ⇸ S
EVENTS
   INITIALISATION   ≙
   STATUS
     ordinary
   BEGIN
     act1   :   t≔Rzero
     act2   :   plantV≔{Rzero↦plantVInit}
   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
   Abstract_Tank_Ctx
EXTENDS
   System_Ctx
CONSTANTS
   V_high
   V_low
   V0
   f_evol_V
AXIOMS
   axm1   :   V_high ∈ RReal
   axm2   :   V_high ↦ V_low ∈ gt
   axm3   :   V_low ∈ RReal
   axm4   :   V_low ↦ Rzero ∈ gt
   axm5   :   V0 ∈ RRealPlus
   axm6   :   f_evol_V ∈ RReal ⟶ (TIME × S ⟶ S)
   axm7   :   ∀ ctrlV · ctrlV ∈ RReal ⇒ (f_evol_V(ctrlV) =
                    (λ t ↦ V · t ∈ TIME ∧ V ∈ RReal | ctrlV))
   axm8   :   V0=plantVInit
END
```

```
MACHINE
   Abstract_Tank_M
REFINES
   System_M
SEES
   Abstract_Tank_Ctx
VARIABLES
   t
   V
INVARIANTS
   inv1   :   V ∈ Closed2Closed(Rzero, t) ⇸ RRealPlus
   inv2   :   V=plantV
EVENTS
   INITIALISATION   ≙
   STATUS
     ordinary
   BEGIN
     act1   :   t≔Rzero
     act2   :   V≔{Rzero↦V0}
   END

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

   Water_behave   ≙
   STATUS
     ordinary
   REFINES
     Plant
   ANY
     V1
     e
   WHERE
     grd1   :   e ∈ DE(S)
     grd2   :   Solvable(Closed2Closed(Rzero, t)\dom(V),e)
                V1 ∈ Closed2Closed(Rzero, t)\dom(V) ⟶ RRealPlus ∧
     grd3   :   AppendSolutionBAP(e,
                Closed2Closed(Rzero, t)\dom(V),
                Closed2Closed(Rzero, t)\dom(V), V1)
   WITH
     plant1   :   plant1=V1
   THEN
     act1   :   V≔V⊲V1
   END

END
```

```
CONTEXT
   Tank_Event_Ctx
EXTENDS
   Abstract_Tank_Ctx
SETS
   EXEC
CONSTANTS
   safeFill
   evt_trigFill
   safeEmp
   evt_trigEmp
   ctrl
   plant
   prg
   f_in
   f_out
   evade_valueFill
   evade_valueEmp
AXIOMS
   axm1   :   partition(EXEC,  {ctrl},{plant},{prg})
   axm2   :   safeFill ∈ (S × RReal) ⟶ BOOL
   axm3   :   evt_trigFill ∈ (S × RReal)×RReal ⟶ BOOL
   axm4   :   safeEmp ∈ (S × RReal) ⟶ BOOL
   axm5   :   evt_trigEmp ∈ (S × RReal)×RReal ⟶ BOOL
   axm6   :   V0 ↦ V_high ∈ lt
   axm7   :   V0 ↦ V_low ∈ gt
   axm8   :   f_in ∈ RReal
   axm9   :   f_in ↦ Rzero ∈ gt
   axm10   :   f_out ∈ RReal
   axm11   :   f_out ↦ Rzero ∈ gt
               safeFill = (λ v↦ctrlV ·
                v ∈ S ∧ ctrlV ∈ RReal |
   axm12   :      bool(v ↦V_high ∈ lt)
               )
               evt_trigFill = (λ v↦t1↦ctrlV ·
                v ∈ S ∧  ctrlV ∈ RReal |
                  bool(
               plus(v ↦ times(ctrlV ↦ t1))
   axm13   :          ↦
                        V_high ∈ leq
                       )
               )
               safeEmp = (λ v↦ctrlV ·
   axm14   :    v ∈ S ∧ ctrlV ∈ RReal |
                  bool(v↦V_low ∈ gt)
               )
               evt_trigEmp = (λ v↦t1↦ctrlV ·
                v ∈ S ∧ t1∈ RReal ∧ ctrlV ∈ RReal |
                  bool(
   axm15   :   plus(v ↦times(ctrlV↦t1))
                           ↦
                           V_low ∈ geq
                        )
               )
   axm16   :   evade_valueFill⊆RReal ∧ evade_valueFill={uminus(f_out)}
   axm17   :   evade_valueEmp⊆RReal ∧ evade_valueEmp={f_in}
END
```

```
MACHINE
   Tank_Event_M
REFINES
   Abstract_Tank_M
SEES
   Tank_Event_Ctx
VARIABLES
   t
   V
   ctrlV
   exec
INVARIANTS
   inv1   :   ctrlV∈{f_in,uminus(f_out)}
   inv2   :   exec ∈ EXEC
   inv3   :   exec≠plant ⇒ dom(V)=Closed2Closed(Rzero, t)
   inv4   :   exec=plant ⇒ t∉dom(V)
   inv5   :   ∀x· x∈ dom(V)⇒ V(x) ↦ V_high ∈ leq ∧ V(x) ↦ V_low ∈ geq
EVENTS
   INITIALISATION   ≙
     extended
   STATUS
     ordinary
   BEGIN
     act1   :   t≔Rzero
     act2   :   V≔{Rzero↦V0}
     act3   :   ctrlV ≔f_in
     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_valueFill ⇒evt_trigFill(V(t)↦minus(t1↦t)↦ctrlV)=TRUE
     grd4   :   ctrlV∉evade_valueEmp ⇒evt_trigEmp(V(t)↦minus(t1↦t)↦ctrlV)=TRUE
   THEN
     act1   :   t≔t1
     act2   :   exec≔plant
   END

   Plant_event_tank   ≙
   STATUS
     ordinary
   REFINES
     Water_behave
   ANY
     V1
   WHERE
     grd1   :   exec=plant
     grd2   :   V1 ∈ Closed2Closed(Rzero, t)\dom(V) ⟶ RRealPlus
     grd3   :   ode(f_evol_V(ctrlV),V1(t),t) ∈ DE(S)
     grd4   :   Solvable(Closed2Closed(Rzero, t)\dom(V),
                             ode(f_evol_V(ctrlV),V1(t),t))
                AppendSolutionBAP(ode(f_evol_V(ctrlV),V1(t),t),
     grd5   :   Closed2Closed(Rzero, t)\dom(V),
                Closed2Closed(Rzero, t)\dom(V), V1)
     grd6   :   ∀xx· xx∈ dom(V1)⇒ V1(xx) ↦ V_high ∈ leq ∧ V1(xx) ↦ V_low ∈ geq
   WITH
```

```
    e   :   e=ode(f_evol_V
            (ctrlV),V1(t),t)
  THEN
    act1   :   V≔V◁V1
    act2   :   exec≔ctrl
  END

  Ctrl_normal   ≙
  STATUS
    ordinary
  ANY
    nCtrlV
  WHERE
    grd1   :   exec=ctrl
    grd2   :   nCtrlV∈{f_in,uminus(f_out)}
    grd3   :   nCtrlV=f_in ⇒ safeFill(V(t)↦ f_in)=TRUE
    grd4   :   nCtrlV=uminus(f_out) ⇒safeEmp(V(t)↦ uminus(f_out))=TRUE
  THEN
    act1   :   exec≔ prg
    act2   :   ctrlV≔nCtrlV
  END

  Ctrl_emptying   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   exec=ctrl
    grd2   :   safeEmp(V(t)↦ uminus(f_out))=TRUE
  THEN
    act1   :   exec≔prg
    act2   :   ctrlV≔uminus(f_out)
  END

  Ctrl_filling   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   exec=ctrl
    grd2   :   safeFill(V(t)↦ f_in)=TRUE
  THEN
    act1   :   exec≔prg
    act2   :   ctrlV≔f_in
  END

END
```

```
CONTEXT
   Tank_Time_Ctx
EXTENDS
   Tank_Event_Ctx
CONSTANTS
   epsilon
   safeEpsilonFill
   safeEpsilonEmp
AXIOMS
   axm1   :   epsilon ∈ TIME ∧ sigma↦epsilon ∈leq
   axm2   :   safeEpsilonFill ∈ (S × RReal) ⟶ BOOL
              safeEpsilonFill = (λ v↦ctrlV · v ∈ S ∧ ctrlV ∈ RReal |
              bool(
                     plus(v↦times(ctrlV↦epsilon))
   axm3   :          ↦
                     V_high ∈ leq
              )
              )
              safeEpsilonEmp = (λ v↦ctrlV · v ∈ S ∧ ctrlV ∈ RReal |
              bool(
                   plus(v↦times(ctrlV↦epsilon))
   axm4   :          ↦
                    V_low ∈ geq
              )
              )
   axm5   :    Rzero↦epsilon∈ lt
END
```

```
MACHINE
   Tank_Time_M
REFINES
   Tank_Event_M
SEES
   Tank_Time_Ctx
   Theorems
VARIABLES
   t
   V
   ctrlV
   exec
INVARIANTS
                 ∃ t1·t1 ∈TIME ∧ dom(V)=Closed2Closed(Rzero,t1) ∧
                     minus(t↦t1)↦epsilon ∈leq ∧
   inv1   :       (exec≠plant  ⇒ t1=t) ∧
                  (exec=plant⇒  t↦t1∈gt) ∧
                  (ctrlV∉evade_valueFill ∧ exec=plant   ⇒ safeEpsilonFill(V(t1)↦ctrlV) = TRUE) ∧
                  (ctrlV∉evade_valueEmp ∧ exec=plant   ⇒ safeEpsilonEmp(V(t1)↦ctrlV) = TRUE)
   inv2   :      ctrlV∉evade_valueFill ∧ exec=prg   ⇒
                 safeEpsilonFill(V(t)↦ctrlV) = TRUE
   inv3   :      ctrlV∉evade_valueEmp ∧  exec=prg    ⇒
                 safeEpsilonEmp(V(t)↦ctrlV) = TRUE
                 ∀t1,t2· t1∈TIME ∧ t2∈TIME ∧
   inv4   :       dom(V)=Closed2Closed(Rzero,t1) ∧ dom(V)=Closed2Closed(Rzero,t2)
                 ⇒
                 t1=t2
EVENTS
   INITIALISATION   ≙
   STATUS
     ordinary
   BEGIN
     act1    :    t≔Rzero
     act2    :    V≔{Rzero↦V0}
     act3    :    ctrlV ≔f_in
     act4    :    exec ≔ ctrl
   END

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

   Plant_time_tank   ≙
   STATUS
     ordinary
   REFINES
     Plant_event_tank
   ANY
     V1
     lastTime
     epsilon1
   WHERE
```

```
  grd1   :   exec = plant
  grd2   :   lastTime∈ TIME ∧ dom(V)=Closed2Closed(Rzero, lastTime)
  grd3   :   t↦lastTime∈ gt ∧ lastTime∈ dom(V)
  grd4   :   epsilon1=minus(t↦lastTime)
             V1=(λ t1 · t1 ∈ RReal ∧ t1↦ lastTime ∈ gt ∧  t1 ↦ t ∈ leq|
                   plus(
                        times(ctrlV ↦ epsilon1)
  grd5   :                ↦
                        V(lastTime)
                       ))
  grd6   :   ode(f_evol_V(ctrlV),V1(t),t) ∈ DE(S)
  grd7   :   Solvable(Closed2Closed(Rzero, t)\dom(V),
                          ode(f_evol_V(ctrlV),V1(t),t))
             solutionOf(
              Closed2Closed(Rzero, t)\dom(V),
  grd8   :   (Closed2Closed(Rzero, t)\dom(V)) ⩤ V1,
             ode(f_evol_V(ctrlV),V1(t),t)
             )
THEN
  act1   :   V≔V⩤V1
  act2   :   exec≔ctrl
END

Ctrl_normal_time   ≙
STATUS
  ordinary
REFINES
  Ctrl_normal
ANY
  nCtrlV
WHERE
  grd1   :   exec=ctrl
  grd2   :   nCtrlV∈ {f_in,uminus(f_out)}
  grd3   :   nCtrlV=f_in⇒safeEpsilonFill(V(t)↦ f_in)=TRUE
  grd4   :   nCtrlV=uminus(f_out)⇒safeEpsilonEmp(V(t)↦ uminus(f_out))=TRUE
THEN
  act1   :   exec≔ prg
  act2   :   ctrlV≔nCtrlV
END

Ctrl_emptying   ≙
STATUS
  ordinary
REFINES
  Ctrl_emptying
WHEN
  grd1   :   exec=ctrl
  grd2   :   safeEpsilonEmp(V(t)↦ uminus(f_out))=TRUE
THEN
  act1   :   exec≔prg
  act2   :   ctrlV≔uminus(f_out)
END

Ctrl_filling   ≙
STATUS
  ordinary
REFINES
  Ctrl_filling
WHEN
  grd1   :   exec=ctrl
  grd2   :   safeEpsilonFill(V(t)↦ f_in)=TRUE
THEN
  act1   :   exec≔prg
  act2   :   ctrlV≔f_in
END
```

END