Modeling Hysteresis in CLIP

David K. Wittenberg and Timothy J. Hickey

Brandeis University

Outline

- Tools
  - Interval Arithmetic
  - Constraint Logic Programming (CLP)
- Hybrid Systems
- Using CLP(F) for Hybrid Systems
  - Modeling
  - Analysis
- Examples

Interval Arithmetic

[Moore, 1966]
Introduced rigor to calculations on floating point numbers

- Closed under + - x / (if interval doesn't contain 0)
- Standard inequalities are all true.
- Distributive law does not hold

Constraint Logic Programming

[Jaffar, Lassez 1987] Logic programming - each result states that there is a proof of the result
Use constraints rather than equalities.
powerful, Fastest method for some problems
good for scheduling - used by sports leagues
Doesn't give answer - just limits places where solution can be

CLP(I)

[Hickey 2000]
CLP over Reals
Careful Rounding in IEEE 754 Floating Point

```
| ?- {X^2=2,X>0}.
X = 1.41421356237309... ? ;
no
| ?-
```

Multiple Solutions

```
| ?- {X^2=2}.
X = [-1.4142135623730953675192267837 7,
       1.4142135623730953675192267837 7] ? ;
no
| ?-
```

Multiple Solutions

```
| ?- {X^2=2}.
X = [-1.4142135623730953675192267837 7,
      1.4142135623730953675192267837 7] ? ;
no
| ?-
```

```
| ?- {X^2=2},solve_clip(queue,[X],0.000001).
X = 1.4142135623730 9... ? ;
X = -1.4142135623730 9... ? ;
(10 ms) no
```

CLP(F) – ODE solver

[Hickey 2000]
Extension of CLP(I)
Uses Taylor Expansions with Remainder

CLP(F) example

$$Q(F, A, E) \equiv$$
$$F \in \mathcal{H}([0,1]), F' = F, F([0,1]) \subseteq [-100, 100],$$
$$F(0) = 1, F(A) = 2, F(1) = E$$

```
| ?- type([F],function(0,1)),
     {[ ddt(F,1)=F,    F in [-100,100],
       eval(F,0)=1,eval(F,A)=2, eval(F,1)=E ]}.

A = .6931471...    E = 2.7182818...
```

Hybrid Systems

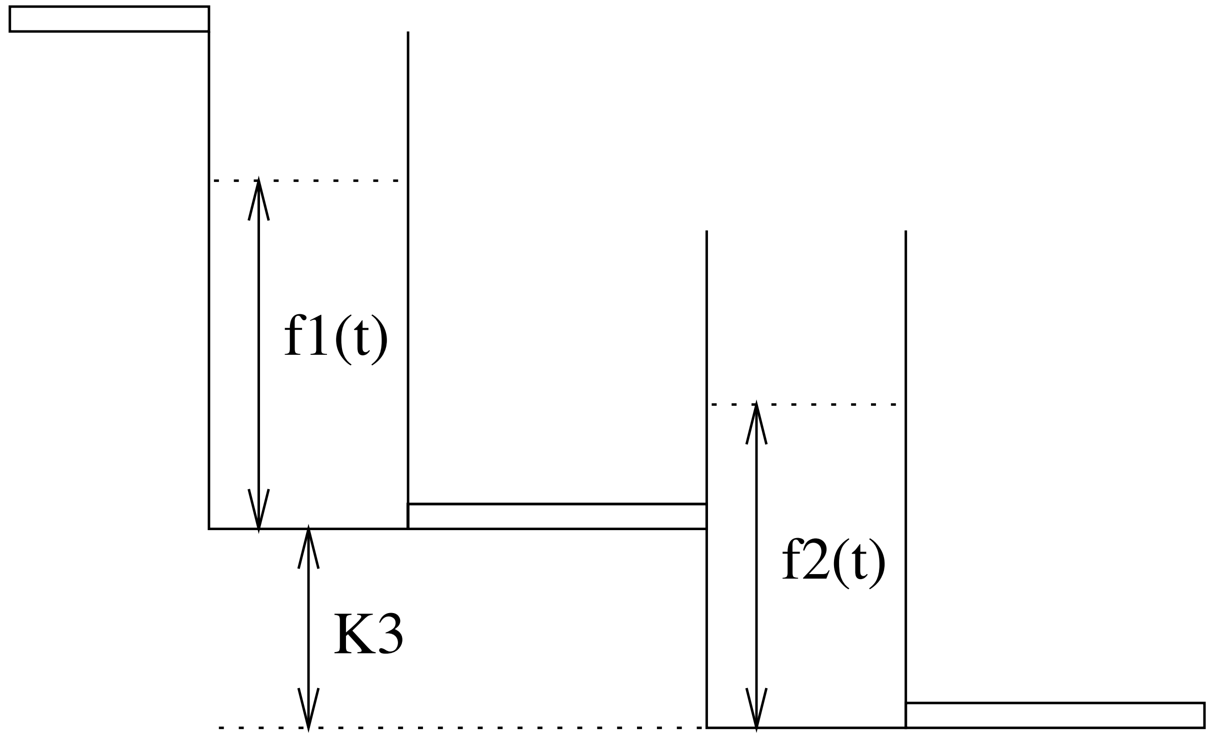
[Fahrland 1970]

- Analog part - changes state according to ODEs
- Digital part - changes state in 0 time

CLP(F) for Hybrid Systems

- Rigorous
- ODEs translate directly to CLP(F)
- Handles non-linear ODEs
- Logical Semantics

Two Tank System



ODEs for Two Tank System

$$f_1' = \begin{cases} k_1 - k_2\sqrt{f_1 - f_2 + k_3} & f_2 > k_3 \\ k_1 - k_2\sqrt{(}f_1) & f_2 \leq k_3 \end{cases}$$

$$f_2' = \begin{cases} k_2\sqrt{f_1 - f_2 + k_3} - k_4\sqrt{f_2} & f_2 > k_3 \\ k_2\sqrt{(}f_1) - k_4\sqrt{(}f_2) & f_2 \leq k_3 \end{cases}$$

Code for Two Tank System

$$f_1' = k_1 - k_2\sqrt{f_1 - f_2 + k_3} \qquad f_2 > k_3$$

$$f_2' = k_2\sqrt{f_1 - f_2 + k_3} - k_4\sqrt{f_2} \quad f_2 > k_3$$

```
twotank(case1,X10,X20,T0,X11,X21,T1,[K1,K2,K3,K4]
   :-      decls([X1,X2],function(T0,T1)),
  {[ ddt(X1,1) = K1 - K2*psqrt(X1-X2+K3),
     ddt(X2,1) = K2*psqrt(X1-X2+K3) - K4*psqrt(X2
     eval(X1,T0)=X10,   eval(X1,T1)=X11,
     eval(X2,T0)=X20,   eval(X2,T1)=X21,
     X1 in [E,1000],  X2 in [K3,1000], E=0.000000
    ]}.
```

Analysis of Hybrid System

If tank levels X_0 *for the upper tank and* Y_0 *for the lower satisfy*

$0.62 \leq X_0 \leq 0.63 \quad \wedge \quad 0.558 \leq Y_0 \leq 0.567$

at time 0, then for all times $n \cdot 0.1$ *the tank levels* X *and* Y *satisfy*

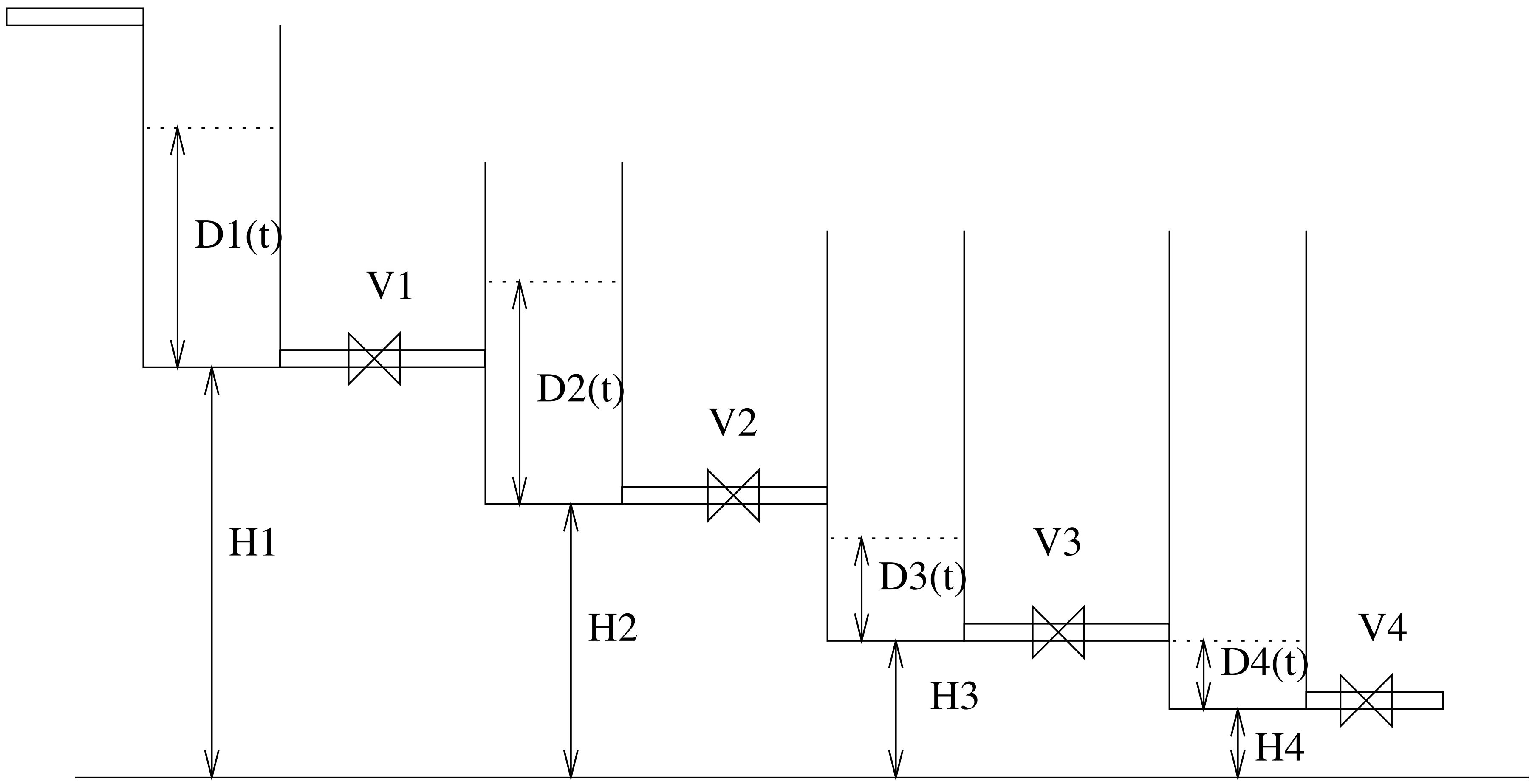
$0.61922 \leq X \leq 0.63083 \wedge 0.55674 \leq Y \leq 0.56815$

```
{X0 = [0.62,0.63],Y0=[0.558,0.567]},
    ks(Ks), twotank(case1,X0,Y0,0.0,X,Y,0.1,Ks)
    solve_clip(fwchk,[X,Y],N).

N=0  X=[.61931, .63069]   Y=[.55697, .56802]
N=1  X=[.61964, .63035]   Y=[.55758, .56741]
N=2  X=[.61985, .63013]   Y=[.55796, .56703]
N=3  X=[.61995, .63004]   Y=[.55812, .56687]
N=4  X=[.62000, .62999]   Y=[.55819, .56680]
```

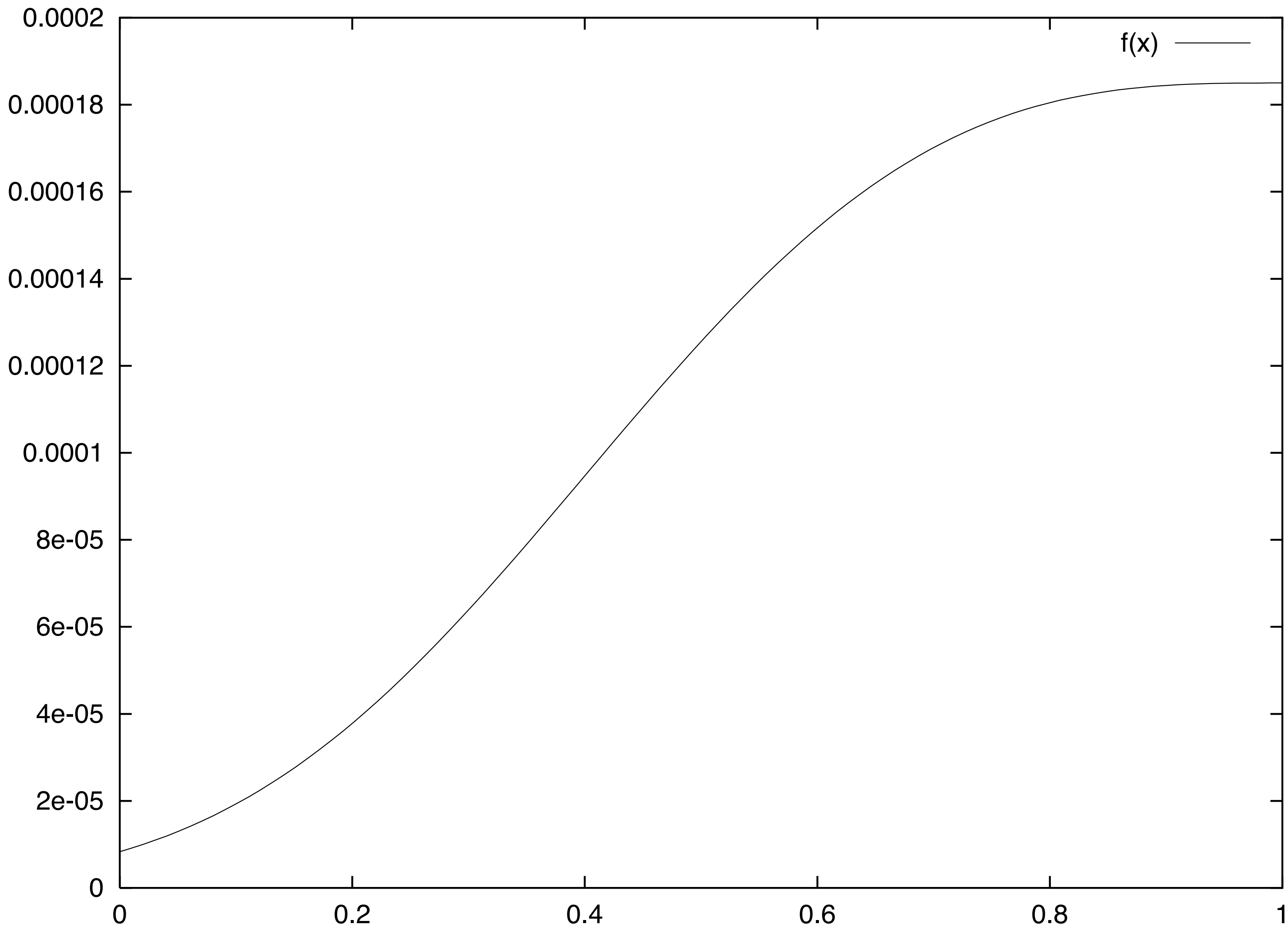
More Elegant Proof of Safety Property

```
| ?- {X10 = [0.62,0.63],X20=[0.558,0.567]},
     ks(Ks),
     twotank(case1,X10,X20,0.0,X11,X21,0.1,Ks),
 ({X11<0.62} ; {X11>0.63};
     {X21<0.558}; {X21>0.567}).

(1330 ms) no
| ?-
```

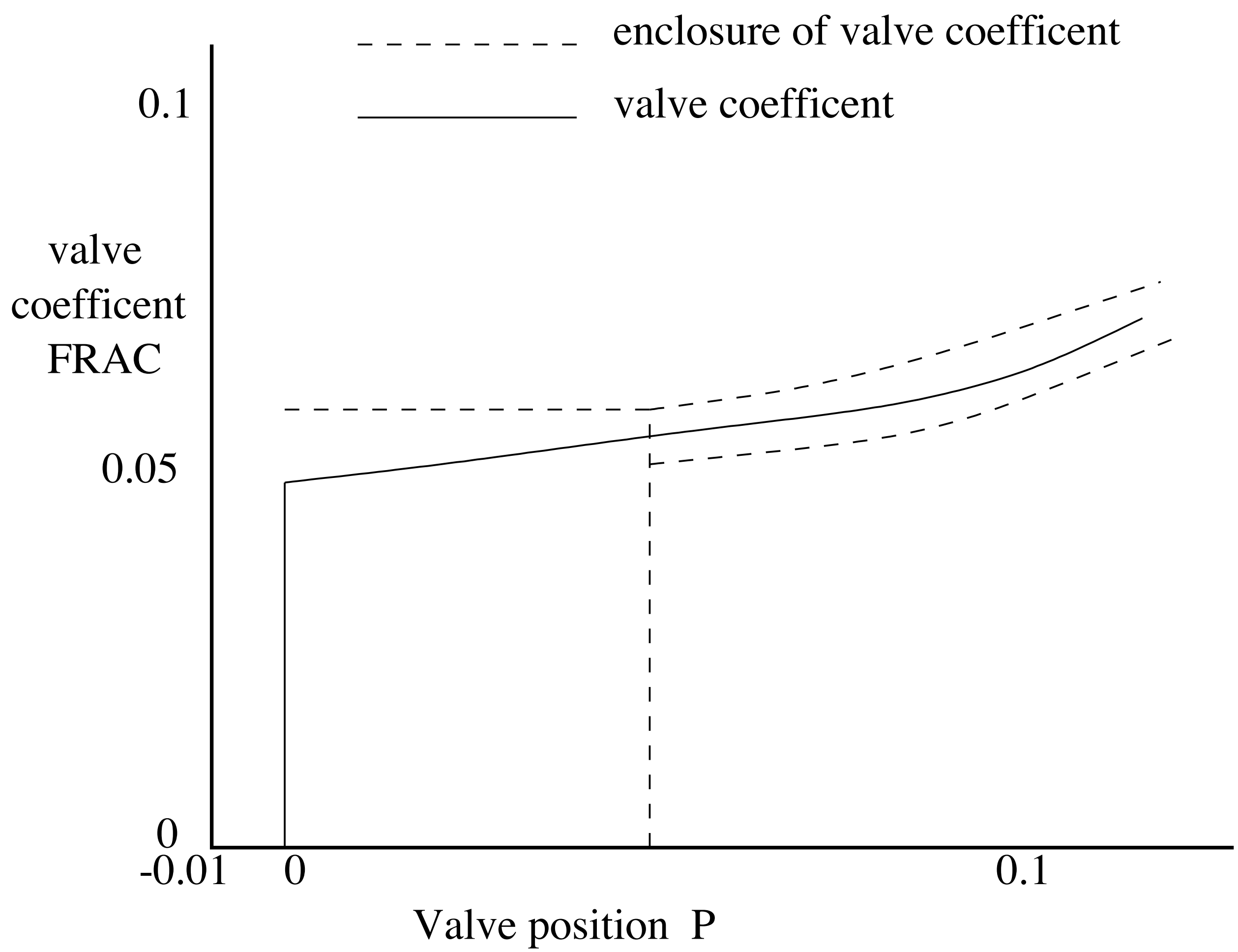
Four Tanks



Valve Behaviour



Enclosing Poor behaviour



Equations For Each Tank

$$
\begin{aligned}
D'_j(t) &= F_{j-1}(t) - F_j(t) \\
R_j(t) &= C_j \cdot e^{E_j \cdot (1 - P_j(t))^3}
\end{aligned}
$$

$$
F_j = \begin{cases} R_j(t)\sqrt{D_j(t) - D_{j+1}(t) + H_{j+1}} & D_{j+1}(t) > H_j - H_{j+1} \\ R_j(t)\sqrt{(D_j)} & D_{j+1}(t) \leq H_j - H_{j+1} \end{cases}
$$

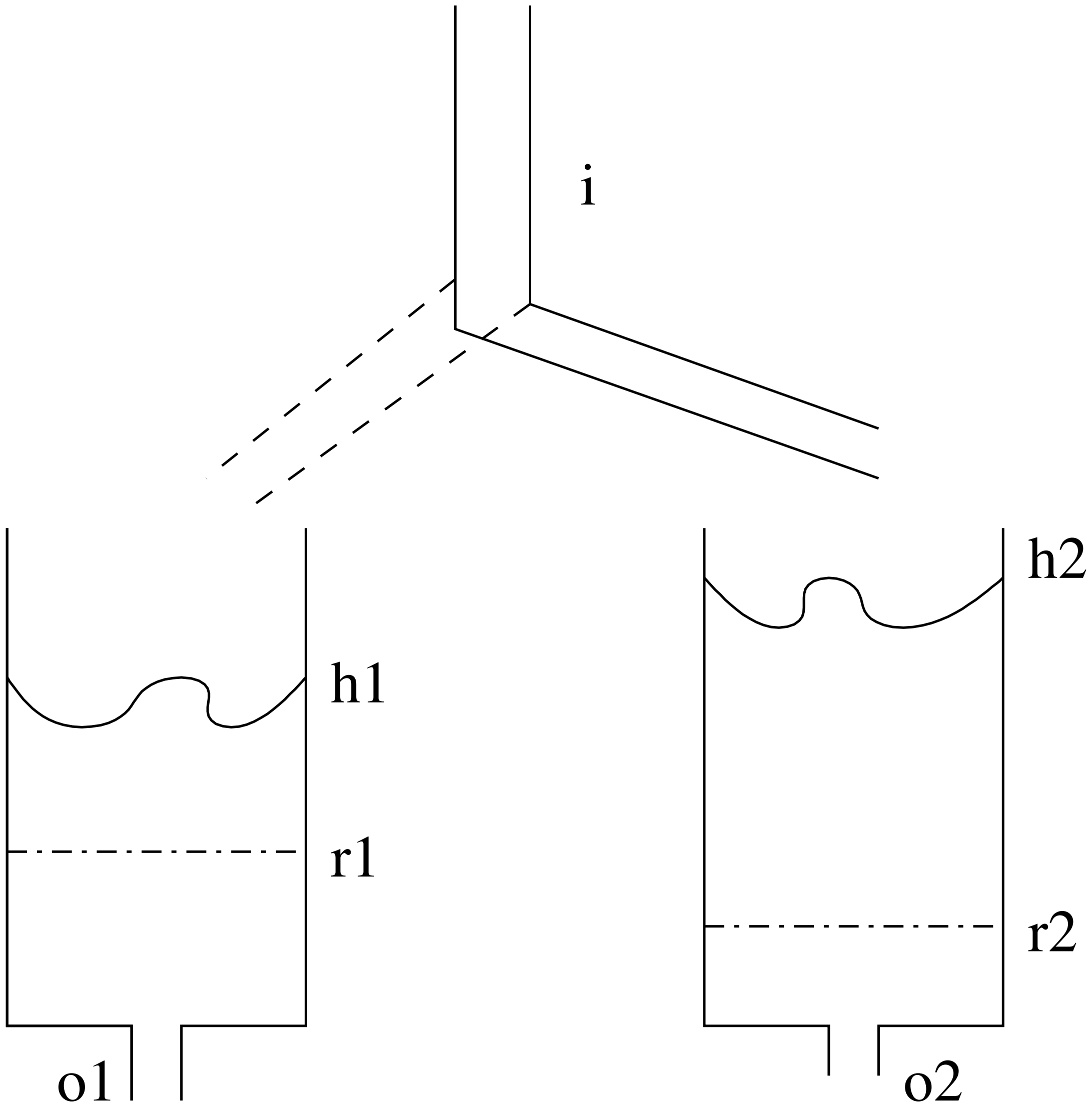
CLP(F) Program For Each Tank

```
middle_tank(above,F1,D2,F2,D3,H2,KC2,FRAC2,H3,_E)
    :-     {[F2=KC2*FRAC2*psqrt(D2-D3+H),
      ddt(D2,1)=F1-F2, H=H2-H3, D3 in [H,1000] ]}

middle_tank(near,F1,D2,F2,D3,H2,KC2,FRAC2,H3,E)
    :-         decls([EF],function(_,_)),
   {[F2=KC2*FRAC2*psqrt(D2-EF), ddt(D2,1)=F1-F2,
    H=H2-H3, D3 in [-2*E+H,2*E+H], EF in [0,2*E] ]

middle_tank(below,F1,D2,F2,D3,H2,KC2,FRAC2,H3,E)
     :-   {[F2=KC2*FRAC2*psqrt(D2),
        ddt(D2,1)=F1-F2, H=H2-H3, D3 in [E,H]]}
```

Helper functions

```
valve_coef(normal,FRAC,P,KE) :-
      {[FRAC=exp(KE*((1-P))**3), FRAC in [0,1],
                                 P in [0.01,1] ]}


valve_state_change(opening,trans,_P_before,
                      opening,normal,P_after) :-
                          {P_after=0.01}.
```

Zeno Behavior



Hysteresis

- Prevent “Chatter”
- Delay in switching states
- Often used in physical systems
- Here used both to reflect physics, and to prevent Zeno behaviour

Contributions of this Work

- Improved Rigor in Dealing with Hybrid Systems
- Improved Ease of Describing Hybrid Systems
- Modeling of Hybrid Systems with Non-Linear ODEs
- Modeling of Points where ODEs are Non-Analytic
- Modeling of Areas with “Messy” or Poorly Understood Physics
- Improved Ease of Analysis

Conclusions

- CLP(F) is a powerful tool
- Simple Semantics – Easier argument that code is correct
- Constraint Approach allows one to work “backward”